Patch-based Graphical Models for Image Restoration

Patchgebaseerde grafische modellen voor beeldrestauratie

Tijana Ružić

Promotoren: prof. dr. ir. W. Philips, prof. dr. ir. A. Pižurica
Onderzoeksgroep TELIN-IPI-iMinds
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen

Vakgroep Telecommunicatie en Informatieverwerking
Voorzitter: prof. dr. ir. H. Bruneel
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2013-2014

Members of the jury

prof. dr. ir. Rik Van de Walle (Ghent University, chairman)
prof. dr. ing. Dieter Fiems (Ghent University, secretary)
prof. dr. ir. Wilfried Philips (Ghent University, supervisor)
prof. dr. ir. Aleksandra Pižurica (Ghent University, supervisor)
prof. dr. Ingrid Daubechies (Duke University)
prof. dr. Ann Dooms (Vrije Universiteit Brussel)
dr. ir. Hiêp Quang Luong (Ghent University)
dr. lic. Ewout Vansteenkiste (Ghent University)
prof. dr. ir. Vladimir Zlokolica (University of Novi Sad)

Affiliations

Research Group for Image Processing and Interpretation (IPI)
iMinds
Department of Telecommunications and Information Processing (TELIN)
Faculty of Engineering and Architecture
Ghent University

Sint-Pietersnieuwstraat 41
B-9000 Ghent
Belgium

Samenvatting

Vandaag de dag zijn digitale beelden en video vrijwel niet meer weg te denken: zowel voor privégebruik als commercieel gebruik, de bewakingsindustrie, de medische sector, de textielindustrie als voor productie-industrie, om er maar een paar te noemen. Daarbij stijgen de verwachtingen van de gebruikers, over de kwaliteit van hun digitaal beeldmateriaal, jaar na jaar. Producenten zijn continu in de weer om die verwachtingen in te lossen, door de beeldvormingtoestellen te verbeteren, bijvoorbeeld door het gebruik van optica en camerasensoren van hoge kwaliteit, wat dan weer leidt tot hogere fabricagekost. Langs de andere kant kan geen enkele hoeveelheid geld de wetten van de fysica opheffen (bijvoorbeeld de diffractielimiet). Daardoor zullen digitale beelden nooit perfect zijn wat betreft resolutie, ruis, scherpte, ...

De beperkte kwaliteit van digitale afbeeldingen wordt niet alleen veroorzaakt door het beeldvormingsproces. Het is een recente trend om oud beeldmateriaal te digitaliseren, bijvoorbeeld oude foto's en films, kunststukken uit musea en galerijen. Dit alles heeft als doel om archivering, bewaring en onderzoek te vergemakkelijken. Deze beelden kunnen van zeer hoge resolutie zijn, in het bijzonder in het geval van gedigitaliseerde schilderijen, omdat dit de toeschouwer in staat stelt om de aller-fijnste details van de schilderkunst te kunnen bewonderen. Jammer genoeg heeft dit soort beeldmateriaal nog last van verdere degradaties, zoals krassen in oude foto's, stof in oud filmmateriaal, vlekken en scheurtjes in schilderijen. Veel van deze fenomenen worden veroorzaakt door veroudering.

Al deze degradaties, ongeacht of ze nu het resultaat zijn van veroudering of imperfecties in de beeldvormingtechniek, kunnen verholpen worden door gebruik te maken van digitale nabewerkingtechnieken. Dit verbetert niet alleen de visuele ervaring, maar maakt het ook makkelijker om analyse te doen van de beeldinhoud in toepassingen zoals bewaking, spooronderzoek, satellietobservatie, medische diagnostiek en kunsthistorische analyse. Sommige van deze digitale nabewerkingtechnieken kunnen ook gebruikt worden om beelden aan te passen, bijvoorbeeld om ongewenste elementen uit een afbeelding te verwijderen. Voorbeelden zijn geprinte datums, watermerken, tekst, logos of zelfs hele objecten (mensen of borden in landschapsfoto's).

Voor deze thesis zijn digitale nabewerkingtechnieken ontwikkeld om beelden te restaureren en te bewerken na acquisitie. Onze nadruk lag op *image inpainting*, of het vervolledigen van een afbeelding, wat het invullen van ontbrekende gebieden, op een visueel aanvaardbare manier, in een afbeelding inhoudt. Op die manier slagen we erin om artefacten die veroorzaakt wor-

den door veroudering of ongewenste elementen in een afbeelding te verwijderen door de ongewenste elementen te beschouwen als ontbrekende gebieden. We ontwikkelden methodes om inpainting toe te passen op zogenaamde natuurlijke afbeeldingen, bijvoorbeeld afbeeldingen van natuurlijke scènes, en voor het verwijderen van krassen in gedigitaliseerde schilderijen. Een andere toepassing die deze thesis beschouwt, is *superresolutie* (SR), dit houdt in dat een hoogresolutie (HR) beeld wordt opgebouwd uit een of meerdere laagresolutie (LR) beelden, met behulp van geschatte hoge frequenties. De belangrijkste aanpakken die onderzocht zijn, zijn graafmodellering, in het bijzonder modelleringen met behulp van Markov Random Fields (MRF), voorstelling die gebruik maken van patches, zelfgelijkenis van afbeeldingen en het gebruik van textuureigenschappen om de beeldcontext te beschrijven.

MRF's worden vaak gebruikt in beeldverwerking- en computervisieproblemen omdat ze een handige en consistente manier bieden om contextuele informatie te modelleren. Deze contextuele afhankelijkheden bestaan onvermijdelijk in afbeeldingen omdat pixels, of andere bouwblokken van een afbeelding, spatiaal gecorreleerd zijn. Daarenboven zijn MRF's in staat om de *globale* context in termen van *locale* interacties te modelleren, wat dit soort modellen elegant en computationeel aantrekkelijk maakt. MRF's worden vaak gebruikt als voorkennis bij Bayesiaanse inferentie, zoals Maximum a posteriori (MAP) estimatie, waarbij het doel is om onbekende afbeeldingattributen te schatten uit de beschikbare data, die soms onvolledig of beschadigd is. De eerste bijdrage die in deze thesis beschreven wordt, is de ontwikkeling van een nieuwe suboptimale *inferentiemethode* voor MAP estimatie met MRF voorkennis. Deze techniek presteert heel goed wanneer toegepast op enorme grafen met veel korte lussen, wat een grote flexibiliteit inhoudt wat betreft het definiëren van spatiale interactie tussen beeldentiteiten. Het kernidee is om informatie te laten propageren doorheen de onderliggende graaf van het MRF model, door middel van het verzenden van een enkelvoudig "consensus"-bericht vanuit de buren naar de centrale node. Vandaar is deze techniek genaamd de "neighbourhood-consensus message passing" (NCMP) techniek. Naast het ontwikkelen van een algemeen raamwerk, hebben we ook een vereenvoudigde versie voorgesteld, die toepasbaar is voor grote omgevingen. Experimentele resultaten voor vier verschillende referentietesten bewijzen het potentieel van de voorgestelde methoden.

Een recente trend in beeldverwerking is het gebruik van patches, dit zijn blokvormige structuren van pixelwaarden, als kenmerken ter beschrijving van de centrale pixel in het blok. Patches worden zelfs gebruikt als atoom bij de synthesetoepassingen zoals textuursynthese, inpainting en superresolutie. In dit geval wordt de MRF gebruikt om voorkennis te coderen over de consistentie van naburige beeldpatches. Volgens dit principe hebben we een nieuwe *single image patch-based super-resolution* methode ontwikkeld. Deze maakt gebruik van het interschaal koppelen van patches en MRF-modellering van een HR afbeelding. De belangrijkste nieuwigheid bij onze methode is dat we de de zelfgelijkenis van beeldpatches in natuurlijke afbeeldingen over ver-

schillende resolutieschalen uitbuiten, eerder dan patches over te nemen uit een externe database. Een andere bijdrage is dat we gebruik maken van onze NCMP-inferentiemethode, om een MAP schatting te krijgen van het onbekende HR beeld. Experimentele resultaten tonen aan dat de voorgestelde methode standaardtechnieken overtreft, terwijl ze visueel beter is of gelijkaardig aan state-of-the-art technieken.

We hebben zelfgelijkenis bij beelden gebruikt in het veld van patchgebaseerde beeldinpainting, door het zoeken naar geschikte kandidaatpatches uit de gekende delen van de afbeelding voor de patch die ontbreekt. De belangrijkste bijdrage is hier de *context-gevoelige* manier van inpainting, die kan gebruikt worden bij elk patchgebaseerd inpaintingsalgoritme. Het idee is om de zoektocht naar geschikte patches te leiden naar interessante gebieden in de afbeelding op basis van contextuele eigenschappen. We bereiken dit door het toekennen van contextbeschrijvingen aan gebieden in de afbeelding. Deze beschrijvingen zijn gebaseerd op textuureigenschappen, verkregen door het convolueren van een afbeelding met spatiale lineaire filters van verschillende oriëntatie- en resolutieschaalgevoeligheid. Voor het ontbrekende gebied in een gegeven regio, worden goed passende kandidaatpatches gevonden uit contextueel gelijkaardige regio's. Het voordeel is tweeledig: de kans dat de gekozen overeenkomsten fout zijn, wordt verminderd en de zoektocht naar goed passende patches wordt zeer versneld, omdat er geen exhaustieve zoektocht over de gehele afbeelding meer nodig is.

We gebruikten deze contextgevoelige aanpak in twee nieuwe inpaintingmethoden: de *greedy block-based context-aware* (GBCA) en de *MRF block-based context-aware* (MBCA) inpainting methoden. In GBCA representeren we de context door gebruik te maken van een combinatie van textuur en kleureigenschappen als contextuele beschrijvingen binnen blokken van vaste grootte. De belangrijkste bijdrage van deze methode is een nieuwe prioriteitsdefinitie gebaseerd op contoureigenschappen, die ook worden geëxtraheerd door de output van lineaire filters van verschillende resolutieschalen en oriëntaties te analyseren. De prioriteit wordt bepaald door de opvulvolgorde, die belangrijk is voor patch-gebaseerde inpainting methoden omdat ze het propageren van beeldstructuren, zoals lijnen of contouren binnen het ontbrekend gebied, bepaalt. In vergelijking met gradientgebaseerde aanpakken, die vaak gebruikt worden in andere patchgebaseerde methoden, behaalt onze prioriteit gebaseerd op contoureigenschappen een betere differentiatie tussen verschillende types van patches en daarmee dus ook een beter uiteindelijk inpaintingsresultaat, zoals wordt aangetoond in de experimentele resultaten.

In de MBCA methode, hebben we onze contextgevoelige aanpak gebruikt om de snelheid en performantie te verhogen van zogenaamde globale patchgebaseerde inpaintingstechnieken met behulp van de MRF voorkennis. Een belangrijke bijdrage van deze methode is de verbeterde voorstellingswijze van context: we onderzochten het gebruik van genormaliseerde texton histogrammen als contextuele beschrijvingen en we introduceerden ook een nieuwe top-down splitsingstechniek, die de afbeelding verdeelt in blokken van variabele

grootte, afhankelijk van de context. We stellen ook een eenvoudige, doch efficiënte manier voor om optimalisatie in een MRF model uit te voeren, door onze NCMP-inferentiemethode uit te breiden zodat ze ook gebruikbaar is voor globale inpaintingsproblemen met een groot aantal labels. We evalueerden deze voorgestelde methode voor twee toepassingen bij wijze van voorbeeld: kras- en tekstverwijdering en bewerking van foto's. De resultaten tonen de kwalitatieve voordelen van de voorgestelde aanpak in vergelijking met de state-of-the-art methodes duidelijk aan. Daarenboven is een snelheidsvoordeel aangetoond in vergelijking met een andere MRF-gebaseerde methode.

Tot slot hebben we een inpaintingsmethode ontwikkeld voor het verwijderen van krassen uit gedigitaliseerde schilderijen. Als een specifiek onderwerp is een unieke gedigitaliseerde versie van het *Lam Gods*, ook wel bekend als het *Gents altaarstuk*. Experimentele resultaten tonen aan dat de voorgestelde methoden andere gerelateerde technieken om krassen te verwijderen overtreffen, terwijl ze nog steeds ruimte laten voor verbetering gezien de specifieke aard van krassen in dit schilderij. Hierdoor hebben we een nieuw patchgebaseerde *krasinpaintingsmethode* ontwikkeld. De voorgestelde methode voert een contextgevoelige inpainting uit, maar eerder dan het beschrijven van de context aan de hand van textuur- en kleureigenschappen, zoals in onze andere methoden, hebben we hier het gebruik van beeldsegmentatie voor contextbeschrijving onderzocht. Naast visuele verbetering, blijkt deze voorgestelde methode een nuttig hulpmiddel voor paleografische analyse van sommige delen van het schilderij, wat zeer interessant is voor kunsthistorische analyse.

In totaal heeft het werk in deze thesis geleid tot 2 tijdschriftpublicaties (waarvan 1 als eerste auteur), 1 ingediend tijdschriftartikel en 2 publicaties als hoofdstuk in een boek (als coauteur). 11 artikels zijn gepubliceerd in internationale en nationale conferenties (waarvan 8 als eerste auteur). 7 abstracts werden gepresenteerd op nationale en internationale conferenties (waarvan 2 als eerste auteur).

Summary

Nowadays, digital images and videos are used virtually everywhere: in private and commercial use, surveillance, medicine, mechanical and textile industry, just to name the few application areas. Moreover, the demands of end users regarding the quality of this digital imaging material are ever increasing. The manufacturers are constantly trying to accommodate these demands by improving the acquisition devices, e.g., by using high precision optics and high-quality camera sensors, which, on the other hand, results in high costs. Furthermore, certain physical limitations of the devices (e.g., diffraction limit) still remain. As a consequence, acquired digital images are imperfect in terms of image resolution, noise, blur, etc.

The insufficient quality of digital images is not only caused by the acquisition process. For example, a recent trend is to digitize old imaging material, e.g., old photographs and films, as well as artwork in museums and galleries, all for the purpose of archiving and dissemination. These images may be of very high resolution, especially in the case of digitized paintings, because in this way the audience is able to appreciate the paintings and their finest details. However, they suffer from other degradations, like scratches in old photographs, dust in old films, stains and cracks in digitized paintings, which are caused by their ageing.

All these degradations, regardless whether they are the result of ageing or using imperfect acquisition devices, can be removed by the means of digital post-processing techniques. This does not only improve the visual experience, but also facilitates the analysis of image content in applications like surveillance, forensics, satellite, medical imaging and art historical analysis. Some of these digital post-processing techniques can also be used for image editing, i.e., altering image content. This can be used to remove unwanted elements from images, for example stamped date, watermarks, text, logos, or even the whole objects (e.g., people or road signs from landscape photos).

In this thesis, we developed digital post-processing techniques to restore and edit images after acquisition. Our main focus is on *image inpainting*, or image completion, which is an image processing task of filling in the missing region in an image in a visually plausible way. In this way, we can remove artefacts caused by ageing and unwanted elements from images by treating them as missing regions. We developed methods for image inpainting both for the so-called natural images, i.e., images of natural scenes, and for crack removal in digitized paintings. Another application that we considered in this thesis is *super-resolution* (SR), which creates a high-resolution (HR) image from one or

more low-resolution (LR) images by estimating missing high frequencies. The main approaches that we explore to achieve these goals are: graphical modelling, in particular Markov random field (MRF) modelling, patch representations and image self-similarity and the use of texture features for describing image context.

MRFs are widely used in image processing and computer vision problems because they provide a convenient and consistent way of modelling contextual constraints. These contextual dependencies inevitably exist in images because image pixels and other image entities are spatially correlated. Furthermore, MRFs are able to model *global* image context in terms of *local* interactions, which makes this model elegant and computationally tractable. MRFs are often used as a prior in problems that involve Bayesian inference, like maximum a posteriori (MAP) estimation, where the goal is to estimate some unknown image attributes from the available image data, which are incomplete or degraded. As the first main contribution of this thesis, we developed a novel suboptimal *inference method* for MAP estimation with the MRF prior, which performs well on huge graphs with many short loops and which allows great flexibility in defining spatial interactions between image entities. The central idea is to propagate information through the underlying graph of the MRF model by sending a single "consensus" message from the neighbourhood to the central node. Hence, we named our method neighbourhood-consensus message passing (NCMP). Besides developing a general framework, we also proposed a simplified version, that is suitable for large neighbourhoods. Experimental results on four different benchmarks show the potentials of the proposed methods.

A recent trend in image processing is to use patches, i.e., square blocks of pixel values, as features describing the central pixel of the patch. Patches are even used as a unit of synthesis in applications like texture synthesis, image inpainting and super-resolution. In that case, MRF can be used to encode prior knowledge about the consistency of neighbouring image patches. Along this line, we developed a novel *single-image patch-based super-resolution method*, which uses cross-scale patch matching and MRF modelling of an HR image. The main novelty of our method is that, instead of using HR patches from an external database, we exploit the self-similarity of image patches in natural images across different scales, thus the HR patches are taken from the input image itself. Another contribution is that we use our NCMP inference method, developed within the course of this research, to obtain the MAP estimate of the unknown HR image. Experimental results show that the proposed method greatly outperforms standard techniques, while being visually better or comparable with state-of-the-art techniques.

We also exploited image self-similarity in the field of patch-based image inpainting, by searching for well-matching candidate patches of the patch to be inpainted in the known part of the image. The main novelty therein is a *context-aware approach* for image inpainting, which can be used with any patch-based inpainting algorithm. The main idea is to guide the search for

patches to the areas of interest based on contextual features. We achieved this by assigning contextual descriptors to image regions, which are based on texture features, obtained by convolving the image with the bank of linear spatial filters at various orientations and scales. For the missing part of the given region, well-matching candidate patches will be found in the contextually similar regions. The benefit is twofold: the chance of choosing wrong matches is reduced and the search for well-matching patches is greatly accelerated (no exhaustive search over the whole known part of the image takes place).

We employed this context-aware approach in two novel inpainting methods: *greedy block-based context-aware (GBCA)* and *MRF block-based context-aware (MBCA) inpainting method*. In the GBCA method, we represented the context by using the combination of texture and colour features as contextual descriptors within image blocks of fixed size. The main contribution of this method is a novel priority definition based on contour features, which are also extracted by analysing filter outputs at various orientations and scales. The priority determines the filling order, which is important for patch-based inpainting methods because it ensures the propagation of image structures, such as lines and contours, inside the missing region. Compared to the gradient-based priority, which is often used in other patch-based methods, our priority based on contour features achieves better differentiation between different types of patches, and hence, better final inpainting result, as demonstrated with experimental results.

In the MBCA method, we employed our context-aware approach to improve the speed and performance of the so-called global patch-based image inpainting with the MRF prior. The important contribution of this method is an improved context representation: we explored the use of normalized texton histograms as contextual descriptors and we introduced a novel top-down splitting procedure, which divides the image into variable-size blocks according to their context. We also proposed a simple and efficient way to perform optimization in the MRF model by extending our NCMP inference method to make it suitable for global inpainting problem with large number of labels. We evaluated the proposed method on two example applications: scratch and text removal and photo-editing. Results demonstrate the benefits of our approach in comparison with state-of-the-art methods in terms of quality and additionally, in comparison with another MRF-based method, in terms of speed.

Finally, we applied the developed inpainting methods for crack removal in digitized paintings. As a case study, we used the digitized versions of the *Adoration of the Mystic Lamb*, also known as the *Ghent Altarpiece*. Experimental results show that the proposed methods outperform related crack inpainting methods, while still leaving some room for improvement due to particularities of cracks in this painting. For that reason, we introduced a novel patch-based *crack inpainting method*. The proposed method performs context-aware inpainting, but rather than describing the context with texture (and colour) features within image blocks of fixed or adaptive sizes, like in our previously proposed methods, we explored the use of image segmentation for context

description. Apart from visual enhancement, the proposed method appears to be a useful tool for paleographical analysis of some parts of the painting, which is of special interest for art historical analysis.

In total, the work conducted during this thesis resulted in 2 journal publications (of which 1 as the first author), 1 journal submission and 2 publications in book chapters (as co-author). 11 papers are published on international and national conferences (of which 8 as the first author). 7 abstracts were presented on international and national conferences (of which 2 as the first author).

Contents

List of Acronyms

2D	Two dimensional
3D	Three dimensional
3DTV	3D television
AFC	Averaged filter outputs and color
ANN	Approximate nearest neighbour
AWGN	Additive white Gaussian noise
BP	Belief propagation
CAD	Controlled anisotropic diffusion
CPU	Central processing unit
FTV	Free viewpoint television
GBCA	Greedy block-based context-aware
GBCA-C	Greedy block-based context-aware with confidence-based priority
GBCA-O	Greedy block-based context-aware with orientation-based priority
GBP	Generalized belief propagation
GC	Graph cut
GPU	Graphical processing unit
GRF	Gibbs random fields
HDMV	Hessian matrix decision value
HDTV	High-definition television
HR	High-resolution
IBP	Iterative back-projection
ICA	Independent component analysis
ICE	Iterated conditional expectations
ICM	Iterated conditional modes
LBP	Loopy belief propagation
LR	Low-resolution
MAP	Maximum a posteriori
MBCA	Markov random field block-based context-aware
MCMC	Markov chain Monte Carlo
MCS	Multiple candidate sparsity-based
MLE	Maximum likelihood estimate
MRF	Markov random field
MSE	Mean squared error
NCMP	Neighbourhood-consensus message passing
NEDI	New edge-directed interpolation
NL	Non-local

p-BP	Priority belief propagation
PBF	Pseudo-Boolean function
PCA	Principal component analysis
PDE	Partial differential equations
PDF	Probability density function
POCS	Projection-onto-convex-sets
PPI	Pixels per inch
PSF	Point spread function
PSNR	Peak signal-to-noise ratio
QPBF	Quadratic pseudo-Boolean function
RAM	Random-access memory
RMSE	Root mean square error
SR	Super-resolution
SSD	Sum of squared differences
SSIM	Structure similarity index
TH	Texton histograms
TV	Total variation
WICM	Weighted iterated conditional modes

1

Introduction

In this thesis, we study and develop graphical patch-based models for the purpose of image restoration and editing. We consider two main applications: image inpainting, i.e., filling in damaged or missing parts of the image and super-resolution, i.e., increasing the image resolution.

1.1 Problem statement

In recent decades, there has been a tremendously growing use of digital images and video due to the wide availability of digital cameras, on the one hand, and the process of digitizing old imaging material, on the other. Furthermore, a recent trend is to digitize artwork in museums and galleries for the purpose of archiving and dissemination.

Although the quality of digital cameras is increasing every day, manufacturing and physical limitations and cost restrictions still limit image quality, in terms of image resolution, noise, etc. Furthermore, some image and video acquisition devices, like web-cameras, cell phones and surveillance cameras, use low-quality sensors, resulting in imaging material of low resolution. A lot of low resolution material was captured with old equipment or in old formats, e.g., NTSC and PAL recordings. Nowadays, this material must be displayed on high-resolution (HR) displays or printed on HR printing devices. Therefore, improving the quality of an image by increasing its resolution has become an important task in image processing.

Digital images obtained by digitizing old imaging material and artwork also suffer from certain artefacts, like scratches in old photographs, dust in old films, stains and cracks in digitized paintings, which represent the signs of their ageing. Removing these artefacts improves the quality of these digitized images and videos, thus improving the visual experience and, in the case of artwork, facilitating art historical and digital image analysis. However, the demands of end users regarding digital images extend beyond quality improvement. In particular, they need to edit the images in order to remove unwanted elements, e.g., stamped date, watermarks, text, logos, or even the whole objects. Removing artefacts and unwanted objects can be achieved with

the image processing technique called image inpainting, or image completion. Image inpainting fills in the missing or damaged region in an image in a visually plausible way.

In this thesis, we develop digital post-processing techniques to restore and edit images after acquisition. In particular, we aim at increasing image resolution and removing artefacts and unwanted objects by developing novel super-resolution and inpainting methods. We apply these techniques on the so-called natural images, i.e., images of natural scenes, but we also consider the application of crack removal in digitized paintings. The main approaches that we explore to achieve these goals are: graphical modelling, in particular Markov random field (MRF) modelling, patch representations and image self-similarity and the use of texture features for describing image context.

1.2 Topical outline

Super-resolution and inpainting belong to the wider group of image processing and computer vision tasks, where the main problem is to estimate some unknown image attributes from the available image data, which are incomplete or degraded. The unknown attributes can be missing pixel values (as is the case in this thesis), but also the noise-free components of noisy image pixels, values of disparities from a stereo pair, segments of the image to which each pixel belongs, etc. This problem is usually referred to as *labelling*: each pixel or group of pixels must be assigned a label representing the desired attribute. Optimal assignment of labels usually involves Bayesian inference, like maximum a posteriori (MAP) estimation, with an MRF prior [Besag 86, Li 95]. Often, this is equivalently formulated as an energy minimization problem. MRF theory provides a convenient and consistent way of modelling contextual dependencies between image pixels or other image features. In particular, the global image context is elegantly expressed in terms of local interactions.

MAP-MRF modelling has been used for decades in numerous image processing and computer vision problems, such as image restoration [Geman 84, Besag 86, Felzenszwalb 04, Roth 05, Raj 05], image inpainting [Sun 05, Komodakis 07], image segmentation [Li 90, Boykov 01a, Rother 04, Kohli 09], texture modelling [Cross 83, Geman 86], edge detection [Torre 86, Chou 90], stereo matching [Barnard 89, Boykov 01b], super-resolution [Freeman 00, Tappen 03, Wang 05], automatic placement of seams in digital photomontages [Agarwala 04] and many others. The interest in these approaches has recently increased due to the emergence of powerful new optimization (inference) algorithms, such as loopy belief propagation (LBP) [Pearl 88, Yedidia 00, Yedidia 05] and graph cut (GC) [Boykov 01b], which in addition led to more accurate results in different applications, e.g., stereo matching [Szeliski 08, Bleyer 11]. In this thesis, we use these models for super-resolution and inpainting, but we also give our contribution in the domain of inference in MRFs, by developing a novel inference method, whose applicability we demonstrate on other applications as well.

The question that can be posed regarding MRF modelling and Bayesian inference in general is how to define prior knowledge, i.e., contextual dependencies between unknown ideal data. The simplest example of modelling these dependencies is the smoothness prior, which enforces spatial smoothness across the image in a uniform way. This was one of the first priors used in image processing in 1970s. Since then, a lot of research has focused on improving image priors. A recent research direction is the use of image *examples*, i.e., training image data, to learn the prior, rather than choosing a mathematical expression that would describe image behaviour [Elad 09]. This raises at least three other questions: how are examples used, how are they represented and where are they taken from? One way to use examples is directly in the reconstruction process, where the graphical model, like the MRF, can help to reason about the global structure, i.e., to treat an image as a whole. Examples in this case are usually represented as image *patches*, where we will use the term patch to refer to a small square block of values, e.g., raw pixel values or features like high frequencies, etc. In general, a patch is a region with arbitrary, perhaps image-dependent shape. These patches can be taken either from an external database of pairs of high-quality images and their corresponding low-quality versions or from the degraded image itself.

Using patches from the input image itself (the so-called self-examples) is possible because (almost) the same patches tend to recur many times within the image, both within the same scale, i.e., within the image of original size, and across different scales, i.e., within resized versions of the image. This property of an image is called *self-similarity* and it was explored in applications like texture synthesis [Efros 99, Wei 00, Ashikhmin 01, Hertzmann 01, Efros 01, Liang 01, Kwatra 03], image denoising [Buades 05, Dabov 07, Goossens 08, Buades 08], inpainting [Criminisi 04, Komodakis 07, Wexler 07, Bugeau 10] and super-resolution [Ebrahimi 07, Glasner 09, Protter 09, Luong 10], leading to state-of-the-art results in these fields. The underlying idea is to estimate the missing or noise-free value of a pixel by considering its neighbourhood, i.e., a patch centred at a pixel, *and* all similar neighbourhoods, in the whole input image. These self-examples thus serve as multiple (noisy) observations of similar image structures. Self-similarity can also be exploited to estimate the whole patch of pixel values, i.e., the missing part of the patch is replaced with the values from similar patches, which are found based on the known part of the patch. The assumption in this case is that most pixels within the patch can be determined by the pixels that have been previously estimated, thus they can be estimated at once based on the similar patches. Furthermore, the computation time is decreased compared with the pixel-by-pixel estimation because the search is performed for a group of pixels rather than for each pixel separately. This approach has been extensively employed in texture synthesis [Efros 01, Liang 01, Kwatra 03] and image inpainting [Criminisi 04, Komodakis 07], usually producing visually better results.

Self-similarity as recurrence of patches within the same scale and across different scales was recently statistically analysed in [Glasner 09], while

in [Zontak 11] a parametric quantification of patch recurrence was derived. In [Zontak 11], it was also demonstrated that the priors learned from internal patch statistics are more powerful than the ones learned from the external database of images, in the sense that better results can be obtained. When analysing statistics across different scales, the recurrence of the patch means that the patch appears "as is" at different scales (without downsampling the patch). This is different than self-similarity across different scales exploited in fractal-based methods for image coding and upscaling [Jacquin 92, Polidori 97], because there it is assumed that parts of an image repeat on an ever-diminishing scale, a property attributed to the geometrical shapes called fractals [Barnsley 88].

In our techniques we developed for inpainting and super-resolution, the unknown attributes to be estimated are whole patches of raw pixel values. In order to estimate them, we exploit image self-similarity, i.e., we consider known or undamaged patches within the input image itself as possible values for unknown patches, and we encode prior knowledge about the spatial consistency between neighbouring image patches by using an MRF model. Since the number of possible values is unmanageably high for any inference method, we explore different approaches for reducing this number. In this respect, the important contribution of this thesis is the context-aware approach for image inpainting, where the main idea is to guide the search for patches to the areas of interest based on contextual features. In this way, we employ the wider context into the patch selection process instead of just observing a small patch and its known pixels. In order to describe the context, we explore the use of texture features.

Extraction of texture features has been widely studied in different image processing and computer vision tasks, such as automated inspection [Conners 83, Jain 90, Orjuela 13], medical image analysis [Chen 89], remote sensing [Rignot 90, Poggi 05], texture and image segmentation [Malik 01, Puzicha 97, Scarpa 09, Arbelaez 11], image retrieval [Puzicha 97], object detection [Rikert 99, Torralba 10], scene classification [Oliva 01], texture classification [Varma 03], surface recognition [Leung 99, Leung 01], analysis of paintings [van der Maaten 10], etc. There are many approaches on how to extract these features, among which *multi-channel filtering* is one of the most popular. This approach analyses the filter outputs of an image obtained by convolving the image with the bank of linear spatial filters at various orientations and scales. We employ different texture features obtained by analysing these filter outputs in our context-aware approach for inpainting.

1.3 Contributions and publications

The main novelties and contributions that resulted from this research are:

- A new *inference method* for MAP estimation with the MRF prior. The central idea is to propagate information through the underlying graph of

the MRF model by sending a single "consensus" message from the neighbourhood to the central node instead of exchanging messages between pairs of neighbouring nodes. Hence, we name our method neighbourhood-consensus message passing (NCMP). The practical algorithm combines the flexibility of the simple iterated conditional modes (ICM) and the message-passing framework of the more powerful LBP. The proposed method is also a generalization of the iterated conditional expectations (ICE) algorithm. We also develop a simplified version of NCMP, called weighted iterated conditional modes (WICM), that is suitable for large neighbourhoods. This work resulted in one journal publication [Ružić 12c] and two conference publications [Ružić 09, Ružić 11c], where the publication in [Ružić 09] was awarded with the Best Poster Award.

- A novel *single-image patch-based super-resolution method*, which uses cross-scale patch matching and MRF modelling of an HR image. In patch-based super-resolution, the unknown image attributes to be estimated are HR patches, and the local measurements based on which these attributes are estimated are low-resolution (LR) patches from the input LR image. Since the estimation of HR patches solely based on local measurements can be ambiguous, an MRF prior is used to enable the global agreement of HR patches in terms of their local agreement. The main novelty of our method is that, instead of using HR patches from an external database, we exploit the self-similarity of image patches in natural images across different scales. Thus the HR patches are taken from the input image itself. To solve the resulting optimization problem, we use our NCMP inference method, also developed within this thesis, to obtain the MAP estimate of the unknown HR image. This work was published in a conference proceedings [Ružić 11b].

- A novel *context-aware image inpainting approach*, which can be used with any patch-based inpainting algorithm. The main idea is to guide the search for patches to the areas of interest based on contextual features. We achieve this by assigning contextual descriptors to image blocks, and for the missing region within a given block, well-matching candidate patches will be found in the contextually similar blocks. The benefit is twofold: 1) the chance of choosing wrong matches is reduced, and 2) the search for well-matching patches is greatly accelerated (no exhaustive search over the whole known part of the image takes place). As a consequence, the inpainting result is improved. This work was published in a conference proceedings [Ružić 12a], and it has laid the basis for other contributions of this thesis, as described next.

- *GBCA, greedy block-based context-aware inpainting method*, which employs a combination of textural and colour features as contextual descriptors within image blocks of fixed size. The main contribution of this method is a novel *priority definition* based on contour features, which are extracted by analysing filter outputs at various orientations and scales.

The priority determines the filling order, which is important for patch-based inpainting methods because it ensures the propagation of image structures, such as lines and contours, inside the missing region. Therefore, the idea is to give the highest priority to patches containing contours, then textured patches and finally patches in flat areas. Compared to the gradient-based priority, which is often used in other patch-based methods, our priority based on contour features achieves better differentiation between these different types of patches and hence better final inpainting result. This work was published in a conference proceedings [Ružić 13c].

- *MBCA, MRF block-based context-aware inpainting method*, which is a novel global patch-based image inpainting method. Similarly to the super-resolution application, MRF encodes prior knowledge about the consistency of neighbouring image patches. We solve the resulting optimization problem with an efficient low-complexity inference method, which builds upon our NCMP inference method to make it suitable for global inpainting problem with large number of labels. Another important contribution of this method is an *improved context representation*: we explore the use of normalized texton histograms as contextual descriptors and we introduce a novel top-down splitting procedure, which divides the image into variable-size blocks according to their context. This work resulted in a journal submission [Ružić 13b] and a conference proceedings [Ružić 12b].

- A novel *inpainting method for the virtual restoration of artwork*, which enables the removal of signs of ageing of a painting, such as cracks. Specifically, we address the problem of crack removal in the digitized versions of the *Adoration of the Mystic Lamb*, also known as the *Ghent Altarpiece*. The proposed crack inpainting method takes the specific problems of cracks in this painting into account by incorporating some of the earlier ideas of context-aware inpainting, but with different context representation based on image segmentation. The Ghent Altarpiece is one the most important Belgian masterpieces known all over the world. We got involved in this research on the initiative of prof. Ingrid Daubechies from the Mathematics Department, Duke University, USA, who brought us into contact with prof. Mark de Mey from the Royal Flemish Academy of Belgium (KVAB), Belgium and prof. Maximiliaan Martens and Emile Gezels from the Department of Art, Music and Theatre Sciences, Ghent University, Belgium. Together with prof. Ann Dooms and ir. Bruno Cornelis from the Vrije Universiteit Brussels, Belgium, we co-operated on this project, aimed at developing image processing tools for art investigation. The work on crack removal was published in one journal paper as the second author [Cornelis 13], one book chapter as a co-author [Pižurica 13], two conference publications [Ružić 11a,Ružić 13a] and two abstracts were presented in international conferences [Ružić 10, Cornelis 11]. This work also contributed to the research on pearl characterization in the Ghent

Altarpiece conducted by ir. Ljiljana Platiša from the IPI group. This research was conducted within the same project and it resulted in several publications listed in Appendix B.

To summarize, the work presented in this thesis and contributions to other people's work resulted in 2 journal publications (of which 1 as the first author), 1 journal submission and 2 publications in book chapters (as a co-author). 11 papers are published on international and national conferences (of which 8 as the first author). 7 abstracts were presented on international and national conferences (of which 2 as the first author). A list of all the publications published during the course of this research can be found in Appendix B, while a selection of the most important publications is given below:

- T. Ružić, A. Pižurica and W. Philips. *Neighbourhood-consensus message passing as a framework for generalized iterated conditional expectations.* Pattern Recognition Letters, vol. 33, pages 309-318, February 2012.

- B. Cornelis, T. Ružić, E. Gezels, A. Dooms, A. Pižurica, L. Platiša, J. Cornelis, M. Martens, M. De Mey and I. Daubechies. *Crack detection and inpainting for virtual restoration of paintings: The case of the Ghent Altarpiece.* Signal Processing, vol. 93, no. 3, pages 605-619, March 2013.

- T. Ružić and A. Pižurica. *Context-aware patch-based image inpainting using Markov random field modelling.* IEEE Trans. on Image Proc. (submitted).

- T. Ružić, B. Cornelis, L. Platiša, A. Pižurica, A. Dooms, W. Philips, M. Martens, M. De Mey and I. Daubechies. *Virtual restoration of the Ghent Altarpiece using crack detection and inpainting.* In Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS), pages 417-428, 2011.

- T. Ružić, H. Q. Luong, A. Pižurica and W. Philips. *Single image example-based super-resolution using cross-scale patch matching and Markov random field modelling.* In M. Kamel and A. Campilho, editeurs, Proceedings of Int. Conf. on Image Analysis and Recognition (ICIAR), pages 11-20, 2011.

- T. Ružić, A. Pižurica and W. Philips. *Neighbourhood-consensus message passing and its potentials in image processing applications.* In J. T. Astola and K. O. Egiazarian, editeurs, Image Processing: Algorithms and Systems IX; Proceedings of SPIE, volume 7870, 2011.

- T. Ružić and A. Pižurica. *Texture and color descriptors as a tool for context-aware patch-based image inpainting.* In Image Processing: Algorithms and Systems X; and Parallel Processing for Imaging Applications II; Proceedings of SPIE, volume 8295, 2012.

- T. Ružić, A. Pižurica and W. Philips. *Markov random field based image inpainting with context-aware label selection.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), pages 1733-1736, 2012.
- T. Ružić, A. Pižurica and W. Philips. *Exploring contour and texture features for context-aware patch-based inpainting.* In Proceedings of Symp. on Signal Processing, Image Processing and Artificial Vision (STSIVA), pages 1-5, 2013.
- T. Ružić and A. Pižurica. *Context-aware image inpainting with application to virtual restoration of old paintings.* In IEICE Information and Communication Technology Forum (ICTF), pages 1-8, 2013.

1.4 Organization of the thesis

In this thesis, we study MRF and patch-based models for two main applications: image super-resolution and inpainting. Each chapter of the thesis therefore consists of a background and theoretical introduction, a review of state-of-the-art methods and our contribution in the field. The thesis is organized as follows.

In Chapter 2, we cover the problem of optimization (inference) in MRF models. We first introduce the basic theory of MRFs, where we focus on inference in MRFs, rather than on the task of modelling itself. Therefore, we review some representative inference methods. We propose a novel suboptimal inference algorithm, NCMP, which allows great flexibility in defining spatial interactions between the unknown variables that are to be estimated with the MRF. We show how our method represents a generalization of the ICE inference method beyond pairwise interactions. We also derive another inference method as a simplification of the more general NCMP. We demonstrate the potentials of the proposed methods with experimental results on four different benchmarks. Good performance of the NCMP method for patch-based MRF models encouraged us to use it for inference in all the methods proposed in this thesis that use MRFs.

Chapter 3 addresses the problem of image upscaling. We start the chapter by introducing the main applications, problems and degradation models of image upscaling in general. Afterwards, we give an extensive overview of representative image upscaling methods, among which patch-based methods are considered to be very promising because they are capable of recovering missing high frequencies in the HR image, i.e., performing super-resolution. Combining the benefits of graphical and patch-based models and exploiting the image self-similarity across different resolution scales, we develop a new single-image patch-based super-resolution method. The proposed method generates an HR image using cross-scale patch matching, patch-based MRF modelling and our inference technique proposed in Chapter 2. We demonstrate the effectiveness of our method on different natural images in comparison with other image upscaling methods.

Chapters 4, 5 and 6 treat the problem of image inpainting. In Chapter 4, we first give a short introduction to the basic methodology for solving the inpainting problem, which is followed by an extensive overview of inpainting methods from literature, with the emphasis on patch-based methods. We propose a general approach for context-aware patch-based image inpainting, where textural descriptors are used to guide and accelerate the search for well-matching patches. This approach can be employed to improve the speed and performance of any (patch-based) inpainting method. In Chapter 4, we employ the proposed context-aware approach within a novel inpainting method, which explores the use of contour features to determine the filling order of the missing region.

Chapter 5 combines the ideas from three previous chapters within a novel image inpainting method. Firstly, we employ an MRF patch-based model for inpainting application. Secondly, we apply the proposed context-aware approach from Chapter 4 to improve the speed and performance of this MRF-based inpainting method. Thirdly, we improve on this approach by proposing a novel top-down splitting procedure that divides the image into blocks of adaptive sizes based on their context. Finally, we solve the resulting optimization problem with the extension of our inference method from Chapter 2.

Chapter 6 focuses on the application of image inpainting for virtual restoration of digitized old paintings, specifically crack removal. We first describe the cracks in old paintings in general and we review the related work on virtual restoration of artwork. We focus in more detail on the case study of the Ghent Altarpiece, by describing the particularities of cracks in this painting and by briefly introducing the crack detection method developed to tackle these particularities. We apply our methods developed in Chapters 4 and 5 to remove cracks in the Ghent Altarpiece. After analysing the results, we identify the remaining problems and deal with them by developing a novel crack inpainting method.

Chapter 7 gives concluding remarks about the work presented in this thesis. We also discuss some possible directions for future work.

2

Inference in MRF models

Markov random fields (MRFs) are widely used in image processing and computer vision problems because they provide a convenient and consistent way of modelling contextual constraints in images. These contextual dependencies inevitably exist in images because image pixels and other image entities are spatially correlated. Furthermore, MRFs are able to model *global* image context in terms of *local* interactions, which makes MRF models elegant and computationally tractable.

In this chapter, we first introduce basic properties of MRFs in Section 2.2. However, we are not concerned with MRF modelling itself, rather with the design of optimization (inference) algorithms. There are numerous inference methods available in literature, some of which we explain in more detail in Section 2.3. Also, an excellent overview and comparison of inference methods is in [Szeliski 08]. In Section 2.4, we propose a novel suboptimal inference algorithm, which allows great flexibility in defining spatial interactions between image entities. We name the proposed method *neighbourhood-consensus message passing* (NCMP). Furthermore, we develop another version of our NCMP method that we call *weighted iterated conditional modes* (WICM), presented in Section 2.4.4. Example applications and performance comparisons are given in Section 2.5, while convergence is addressed in Section 2.6. Finally, the concluding remarks are given in Section 2.7.

2.1 Introduction

A typical problem in image processing and computer vision consists of estimating some unknown image attributes from the available image data, which are incomplete or degraded. The unknown attributes can be the noise-free components of the noisy image pixels, values of disparities from a stereo pair, missing pixel values, segments of the image to which each pixel belongs, etc. This problem is usually referred to as *labelling*: each pixel or group of pixels is assigned a label representing the desired attribute.

Since the labelling solely based on the available image data can be ambiguous, prior knowledge about spatial context in the image can be addi-

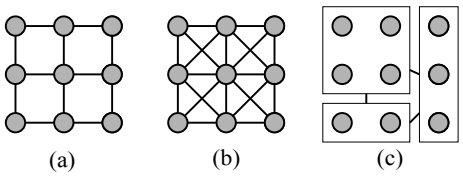


Figure 2.1: Example graphs for MRF models in computer vision and image processing: (a) 4-connected lattice, (b) 8-connected lattice, and (c) irregular lattice (a node corresponds to a group of pixels).

tionally introduced. The spatial constraints in images result from the fact that the value of a pixel (or other image entity) is highly dependent on the values of surrounding pixels. The simplest assumption regarding the spatial context is that the image is smooth everywhere, thus neighbouring pixels are more likely to have similar values. MRF theory provides a convenient and consistent way of modelling these (and more complex) spatial constraints between image pixels or other image features. In particular, the global image context (as a joint probability) is elegantly expressed in terms of local interactions. The resulting labelling problem then involves Bayesian inference, like maximum a posteriori (MAP) estimation, with an MRF prior [Besag 86, Li 95] and it is often formulated as an energy minimization problem.

The so-called MAP-MRF modelling has been used for decades in numerous image processing and computer vision problems, such as image restoration [Geman 84, Besag 86, Felzenszwalb 04, Roth 05, Raj 05, Komodakis 07], image segmentation [Li 90, Boykov 01a, Rother 04, Kohli 09], texture modelling [Cross 83, Geman 86], edge detection [Torre 86, Chou 90], stereo matching [Barnard 89, Boykov 01b], super-resolution [Freeman 00], automatic placement of seams in digital photomontages [Agarwala 04] and many others. The interest in these approaches has recently increased due to the powerful new optimization algorithms, such as loopy belief propagation (LBP) [Pearl 88, Yedidia 00, Yedidia 05] and graph cut (GC) [Boykov 01b], which in addition led to more accurate results in different applications, e.g., stereo matching [Szeliski 08, Bleyer 11].

An MRF is built over a single undirected graph consisting of nodes and edges connecting those nodes. Such structure corresponds to an image, where a node corresponds to a pixel or to a group of pixels (e.g., an image patch) and the edges connecting the nodes represent context dependencies between these image entities. The undirected graphs corresponding to MRF models are depicted in Fig. 2.1. They are mostly lattice-like, but can also be irregular, in the case of labelling more abstract image features, such as corners and lines [Li 95]. These types of graphs contain many loops, in the sense that one node is connected to itself via edges and other nodes. Graphs without loops also exist, like chains and trees, but those are associated with different statistical models.

2.2 Markov random fields

In this section, we introduce the basic theoretical concepts of the MRF model and notations. For more details, see e.g., [Li 95, Winkler 95, Pižurica 02a, Li 09].

2.2.1 Notation and definitions

In this thesis, we consider an underlying MRF model to be an undirected graph represented by a regular rectangular lattice S (Figs. 2.1(a) and (b)). Positions on this lattice, called *nodes*, are represented by a single index $i = 1, \ldots, N_n$ (assuming raster-scan order). The index corresponds to the positions of image pixels or patches. In a graphical *model*, a set of nodes is associated with a family of random variables $\mathbf{X} = (X_1, \ldots, X_n)$ on a set S, where each random variable can take one of L values from the discrete set $x_i \in \Lambda = \{1, \ldots, L\}$.[1] These values are usually referred to as *labels* of a node. Note that in some patch-based models, which we will consider later in this thesis, this label set can be node-specific, i.e., $x_i \in \Lambda_i = \{x_{i_1}, \ldots, x_{i_L}\}$, but for now we will assume it is the same for all nodes. Furthermore, in patch-based models, labels represent patches of pixel values.

We use the notation $X_i = x_i$ to denote the event that the random variable X_i takes the value x_i, and we will refer to this event as the assignment of the label x_i to the node i. The notation $\mathbf{X} = \mathbf{x}$ will be used to abbreviate the joint event $(X_1 = x_1, \ldots, X_{N_n} = x_{N_n})$ and the probability $P(\mathbf{X} = \mathbf{x})$ of the event $\mathbf{X} = \mathbf{x}$ is further abbreviated for simplicity to $P(\mathbf{x})$. Therefore, $\mathbf{x}$ is one possible realization of the random vector $\mathbf{X}$. The family of random variables $\mathbf{X}$ is called a random field, where all its possible realizations have strictly positive probability. Furthermore, we will denote by $\mathbf{X}_A$ a vector of random variables with indices in the set $A \subset S$, and its corresponding realization by $\mathbf{x}_A$.

A random field $\mathbf{X}$ is an MRF if it satisfies the Markov property

$$P(x_i|\mathbf{x}_{S\backslash i}) = P(x_i|\mathbf{x}_{\partial i}), \tag{2.1}$$

where $S\backslash i$ is the set of all nodes except the node i and ∂i is the neighbourhood of i. The Markov property implies that the probability of a node's label conditioned on all other labels reduces to the label's probability conditioned on its *neighbours* only. The distant labels have no influence on the label's probability provided that its immediate neighbours are specified. In simple words, the Markov property allows long-range statistical dependencies to be implicitly described by short-range connections within a specified neighbourhood.

The neighbouring relations are defined formally as follows. The neighbourhood system for the set S is defined as

$$\partial = \{\partial i | \forall i \in S\}, \tag{2.2}$$

[1] In general, this set can also be continuous, but in this thesis we are considering only discrete one because we are working with discrete MRFs.

where ∂i represents the *neighbourhood* and its nodes $k \in \partial i$ represent *neighbours* of i. The neighbourhood must satisfy the following conditions:

1. a node is not neighbouring itself: $i \notin \partial i$
2. the neighbouring relationship is symmetrical: $i \in \partial j \Leftrightarrow j \in \partial i$.

The neighbourhoods most often used in image processing are the first-order (four nearest nodes), corresponding to Fig. 2.1(a), and the second-order neighbourhoods (eight nearest nodes), corresponding to Fig. 2.1(b). MRFs with bigger neighbourhoods are referred to as *highly-connected* MRFs.

2.2.2 Gibbs distribution

A Gibbs random field (GRF) [Malyshev 91] is used in statistical mechanics as a probability model for fluctuations of large physical systems around their equilibrium state. Due to its equivalence with an MRF, it has been used in many applications outside physics. While an MRF is characterized by local spatial interactions, a GRF provides a model for global context since its probability distribution is defined over all nodes in the graph [Pižurica 02a]. A GRF is a random field $\mathbf{X}$ whose realizations $\mathbf{x}$ obey a Gibbs distribution

$$P(\mathbf{x}) = \frac{1}{Z} \exp\Big(-\frac{1}{T} E(\mathbf{x}) \Big), \tag{2.3}$$

where T is the temperature, $E(\mathbf{x})$ is the energy function and $Z = \sum_{\mathbf{x} \in \mathcal{X}} \exp\big(-E(\mathbf{x})/T\big)$ is the normalizing constant called the partition function. Evaluation of Z can be computationally prohibitive because it requires summation over all possible realizations in $\mathcal{X}$. It can be avoided in maximum-probability-based MRF vision models when there are no unknown parameters in the energy function [Li 95].

$P(\mathbf{x})$ is the probability that the realization $\mathbf{x}$ occurs. According to Eq. (2.3), the more probable realizations are the ones with lower energies. The temperature T controls the peakedness of the distribution: when T is high, realizations are almost equally probable, while for low values the distribution concentrates around the global energy minimum.

The equivalence of a Gibbs and a Markov random field is established with the Hammersley-Cilfford theorem, which states that $\mathbf{X}$ is an MRF on S with respect to the neighbourhood system ∂ if and only if $\mathbf{X}$ is a GRF on S with respect to ∂. In other words, if the energy function of a GRF can be expressed as the sum of clique potentials, $E(\mathbf{x}) = \sum_{C \in \mathcal{C}} V_C(\mathbf{x}_C)$, then this GRF is equivalent to an MRF. Different proofs of this theorem can be found in, e.g., [Li 95, Besag 74]. The *clique* C represents a set of nodes, which are all neighbours of one another (see Fig. 2.2 for examples), while $\mathcal{C}$ denotes the set of all possible cliques. $V_C(\mathbf{x}_C)$ is the *clique potential*, which is defined as a function of labels of nodes belonging to C.

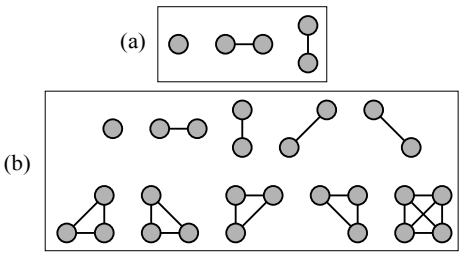


Figure 2.2: Types of cliques for (a) the first-order neighbourhood, and (b) the second-order neighbourhood (bottom row corresponds to clique types of higher-order MRFs).

This theorem enables us to express the joint probability of an MRF, which is a measure of a global context, in a simple way by specifying only local spatial interactions through clique potentials:

$$P(\mathbf{x}) = \frac{1}{Z} \exp\Big(-\frac{1}{T} \sum_{C \in \mathcal{C}} V_C(\mathbf{x}_C) \Big). \tag{2.4}$$

Clique potentials are chosen in practice to favour certain local spatial dependencies, e.g., to encourage smoothness. In this way, the prior knowledge about the image is encoded in the model. In most cases in image processing, the clique potentials are the same for all cliques of a given type, regardless of their spatial position within the lattice, which facilitates computation. Such MRFs are called *homogeneous*. MRF is *isotropic* if clique potential is independent of orientation of the clique. The choice of isotropic MRF is application-dependent and in fact, later in this chapter, we will also make use of an anisotropic MRF.

2.2.3 Common MRF models

In addition to the already mentioned general divisions of MRFs into homogeneous and inhomogeneous or isotropic and anisotropic, in this subsection we will introduce some of the common discrete MRF models that we use in the remaining of this chapter. The most commonly used MRF models are the so-called pairwise MRFs, where cliques consist of pairs of neighbouring nodes. The energy function of such an MRF is

$$E(\mathbf{x}) = \sum_{\langle i,j \rangle} V_{ij}(x_i, x_j), \tag{2.5}$$

where $V_{ij}(x_i, x_j)$ is the pairwise potential representing the interaction of labels of neighbouring nodes. Here, we consider a general form of the pairwise potential, regardless of the homogeneity of the MRF. In the case of a homogeneous MRF, we will denote the pairwise potential as $V(x_i, x_j)$. If cliques consist of more than two neighbouring nodes (see the bottom row of Fig. 2.2(b)), then the

clique potential is a function of more than two labels, and those MRF models are usually referred to as *higher-order* MRFs.

The basic pairwise MRF model is called the *Ising* model [Li 95, Winkler 95], originating from statistical physics, where it was used to model the behaviour of ferromagnets. The main properties of this model are the following:

- the labels are Boolean (binary) variables, e.g., $x_i \in \Lambda = \{-1, 1\}$;[2]
- the neighbourhood is of the first-order (Fig. 2.1(a));
- the model is homogeneous and isotropic;
- the pairwise potential takes the form $V(x_i, x_j) = -\gamma x_i x_j$.

Such a pairwise potential favours assigning the same label to neighbouring nodes for $\gamma > 0$, i.e., $x_i = x_k$ is more probable than $x_i \neq x_k$, while it is the opposite for $\gamma < 0$.

The energy function of the Ising model is $E(\mathbf{x}) = -\gamma \sum_{\langle i,j \rangle} x_i x_j$. It falls into the category of *pseudo-Boolean functions* (PBFs), because the input is Boolean, and the output is real valued, i.e., not Boolean. PBFs are important because they can be optimized in polynomial time if they satisfy the submodularity condition [Kolmogorov 04]

$$f(\mathbf{x}') + f(\mathbf{x}'') \geq f(\mathbf{x}' \vee \mathbf{x}'') + f(\mathbf{x}' \wedge \mathbf{x}''), \tag{2.6}$$

for all label assignments $\mathbf{x}'$ and $\mathbf{x}''$, where $\vee$ and $\wedge$ are component-wise OR and AND, respectively. The pairwise potential of the Ising model satisfies this condition, i.e., it is submodular, because $V(1,-1) + V(-1,1) \geq V(1,1) + V(-1,-1)$, and since the set of submodular functions is closed under addition, also the MRF energy $E(\mathbf{x})$ of the Ising model is submodular [Rother 07, Blake 11]. Graph cut (Section 2.3.4) represents an efficient optimization algorithm for optimizing a subclass of submodular functions, which are common in image processing and computer vision problems.

Another important model is the *Potts* model [Potts 52], which represents a generalization of the Ising model to the problems with multiple labels ($L > 2$). The pairwise potential is defined as

$$V(x_i, x_j) = \begin{cases} \gamma, & \text{if} \quad x_i \neq x_j \\ 0, & \text{if} \quad x_i = x_j. \end{cases} \tag{2.7}$$

The Potts model penalizes any pair of different labels equally with γ ($\gamma > 0$), regardless of the magnitude of the difference.[3] Therefore, it represents a type of a smoothness prior. Some other smoothness priors can be expressed in

[2] Equivalently, binary variables can be $x_i \in \Lambda = \{0, 1\}$.

[3] In the definition of the Potts model in [Won 04], $V(x_i, x_j) = -\gamma$ if $x_i = x_j$, in which case the parallel can be made with the multi-level logistic (MLL) model [Li 95]. In particular, the MLL model represents a generalization of the Potts model for the case of different clique potentials for each clique type.

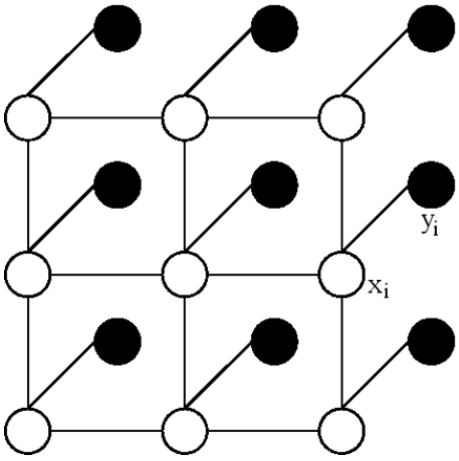


Figure 2.3: Square lattice of nodes, where each node i is associated with an observation y_i (black circles) and a hidden random variable X_i and its realization (label) x_i (white circles). A pairwise MRF with the first-order neighbourhood is imposed on the hidden variables. The edges indicate pairwise cliques.

the general form $V(x_i, x_j) = \min(|x_i - x_j|^r, V_{max})$, and they include, e.g., the truncated linear prior ($r = 1$) and the truncated quadratic prior ($r = 2$) [Szeliski 08]. The choice of the model and the prior highly depends on the application. Some example applications are illustrated later in Section 2.5, while a more comprehensive overview of different applications can be found, e.g., in [Szeliski 08, Blake 11].

2.2.4 Bayesian inference in MRF models

A common problem in image processing and computer vision is to infer unknown variables from the available measurements or observations (evidence assignments). For example, in image denoising, one knows the value of the noisy pixel and aims at inferring the value of the noise-free pixel based on the available noisy value and on the prior knowledge about the image. This prior knowledge can be encoded with an MRF.

The example of such an MRF model with the first-order neighbourhood is sketched in Fig. 2.3. Nodes are associated with the unknown (hidden) random variables defined in Section 2.2.1 (white circles), *and* observations (black circles). Set of observations on S is interpreted as a realization $\mathbf{y} = (y_1, \ldots, y_{N_n})$ of a random vector $\mathbf{Y} = (Y_1, \ldots, Y_{N_n})$ and it represents given image data, such as colour values of image pixels or patches. The same notational conventions apply as for unknown random variables in Section 2.2.1: the joint event $(Y_1 = y_1, \ldots, Y_{N_n} = y_{N_n})$ is abbreviated as $\mathbf{Y} = \mathbf{y}$ and $P(\mathbf{Y} = \mathbf{y})$ as $P(\mathbf{y})$.

Defining such a model naturally leads to an *inference* problem, where the goal is to estimate the underlying properties $\hat{\mathbf{x}}$ given the observed data $\mathbf{y}$, and a certain optimization criterion. Often, the criterion is to maximize the posterior distribution for the possible labels $\mathbf{x}$, given the observations $\mathbf{y}$, $P(\mathbf{x}|\mathbf{y})$, i.e., to compute the MAP estimate

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} P(\mathbf{x}|\mathbf{y}). \tag{2.8}$$

Using the Bayes rule

$$P(\mathbf{x}|\mathbf{y}) = \frac{P(\mathbf{y}|\mathbf{x})P(\mathbf{x})}{P(\mathbf{y})}, \tag{2.9}$$

and if $P(\mathbf{y})$ is independent of $\mathbf{x}$, the MAP estimate can be expressed as

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} P(\mathbf{y}|\mathbf{x})P(\mathbf{x}). \tag{2.10}$$

$P(\mathbf{x})$ is the prior joint distribution of the nodes' labels, encoded by an MRF and indicated by edges in the graph in Fig. 2.3, which are connecting the nodes. This underlying MRF model can take, for example, any of the forms described in Section 2.2.3. The term $P(\mathbf{y}|\mathbf{x})$ is called the *likelihood* of the observations and it describes the relationship between a label and an observation at a node, indicated by an edge between them in Fig. 2.3. It is determined by the knowledge of the reconstruction mechanism or by learning from training data. The likelihood is commonly approximated by assuming conditional independence between random variables $Y_1, \ldots, Y_n$, given the labels $\mathbf{x}$ [Besag 86], or formally

$$P(\mathbf{y}|\mathbf{x}) = P(y_1|x_1) \ldots P(y_n|x_n). \tag{2.11}$$

This aspect is depicted in Fig. 2.3 by the absence of edges between observations themselves. Furthermore, it is assumed that the conditional density functions are the same for each y_i and dependent only on x_i.

If the prior distribution $P(\mathbf{x})$ is a Gibbs distribution (see Eq. (2.4) and the discussion in Section 2.2.2), then it can be shown [Pižurica 02a] that also posterior MRF probability is a Gibbs distribution with posterior energy $E(\mathbf{x}|\mathbf{y})$:

$$P(\mathbf{x}|\mathbf{y}) \propto \exp\Big(- E(\mathbf{x}|\mathbf{y})\Big), \tag{2.12}$$

where $\propto$ denotes proportionality. Now the MAP estimation problem from Eq. (2.8) becomes an energy minimization problem: $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} E(\mathbf{x}|\mathbf{y})$. The posterior energy in general case is represented as

$$E(\mathbf{x}|\mathbf{y}) = \sum_{C \in \mathcal{C}} V_C(\mathbf{x}_C) + \sum_i D_i(x_i, y_i), \tag{2.13}$$

where $V_C(\mathbf{x}_C)$ is the clique potential and $D_i(x_i, y_i)$ is the so-called likelihood energy or data term, which modifies the energy function in order to include

the observations. Note that in general the model is characterized by certain parameters ω, which means that the posterior probability, posterior energy and partition function are dependent on those parameters. However, we will assume that the parameters are given, in which case the posterior energy is fully specified, the MAP-MRF problem is completely defined, and the computation of Z, which is often intractable, is not required.

The posterior energy of the *pairwise* MRF model, introduced earlier in Section 2.2.3, and now including observations, is

$$E(\mathbf{x}|\mathbf{y}) = \sum_{\langle i,j \rangle} V_{ij}(x_i, x_j) + \sum_i D_i(x_i, y_i). \tag{2.14}$$

In belief propagation literature [Yedidia 00,Freeman 00], it is common to consider the joint distribution between observations $\mathbf{y}$ and labels $\mathbf{x}$, which can be written in a factorized form as

$$P(\mathbf{x}, \mathbf{y}) = \prod_{\langle i,j \rangle} \psi_{ij}(x_i, x_j) \prod_i \phi_i(x_i, y_i), \tag{2.15}$$

where $\psi_{ij}(x_i, x_j) \propto \exp(-V_{ij}(x_i, x_j))$ denotes the statistical dependency between labels of pairs of neighbouring nodes and $\phi_i(x_i, y_i) \propto \exp(-D_i(x_i, y_i))$ is the local evidence which models the relationship between an observation and a label. Essentially, $\phi_i(x_i, y_i) = P(y_i|x_i)$. $V_{ij}(x_i, x_j)$ and $D_i(x_i, y_i)$ are the terms from Eq. (2.14).

2.3 Inference methods

Inference methods aim at finding the optimal solution to the problem expressed by the means of the objective function. In image processing and computer vision, the objective function is formulated in terms of given observations, e.g., pixel intensities, and spatial interactions between those pixels, which are derived from the prior knowledge about the image and encoded via MRF. The most popular choice of optimization strategy is MAP estimation (Eq. (2.8)), where the objective function is the posterior probability of the underlying image $\mathbf{x}$ given the observed data $\mathbf{y}$ from Eq. (2.9). This is equivalent to minimizing the energy from Eq. (2.13). Since in this thesis we consider only discrete label set, the inference problem actually represents a combinatorial one.

In this chapter, we are not concerned with how faithfully the objective function models the reality. We are rather focused on the optimization algorithms, i.e., how to retrieve the optimal solution under certain optimization criterion when the objective function is already given.

There are two main issues regarding optimization [Li 95]:

- dealing with the existence of local optima when the objective function is non-convex and
- the efficiency of the algorithm in terms of memory and computation time.

In MAP-MRF labelling problems, computation is often exhaustive (meaning all possible options are examined) or even intractable due to a large number of variables and the loopy structure of the graph. Therefore, it is difficult to respect both above mentioned demands simultaneously. Some methods perform local optimization, which is efficient, but can only provide a local optimum and the solution typically depends on the initial estimate. These combinatorial methods include relaxation labelling [Rosenfeld 76], iterated conditional modes (ICM) [Besag 86], highest confidence first [Chou 90], dynamic programming [Bellman 62], as some of the classical methods listed in [Li 95], and more recent LBP [Pearl 88, Yedidia 01a]. On the other hand, there exist also global methods, like random search methods [Metropolis 53, Geman 84], often combined with simulated annealing [Kirkpatrick 83,Cerny 85], which aim at finding a global optimum. Also GC [Greig 89, Boykov 01b, Boykov 04] and sequential tree-reweighted message passing [Kolmogorov 06] belong to this group under certain conditions. Finding a global optimum is a non-trivial problem when the energy function or joint probability have many local optima, which typically results in exhaustive search. In the remaining of this section, we will review in more detail the inference methods, which are the most relevant for our work.

2.3.1 MCMC samplers and simulated annealing

For most types of problems, there is no efficient algorithm which guarantees to find global optimal solution. In that case, one can settle for an approximate solution at a smaller computational cost. Approximate methods that provide such solutions are, among others, random search methods, such as Markov chain Monte Carlo (MCMC) samplers, which are also slow compared to the more recent methods, but find an optimal solution with high probability. MCMC samplers obtain a sequence of random samples (e.g., a sequence of realizations of an MRF) from a probability distribution for which direct sampling is difficult, i.e., the samples are only approximately from the target distribution. The next sample in the sequence is randomly generated by using the previous sample, thus the sequence represents a Markov chain, and it is used to perform inference, e.g., by estimating marginals as a fraction of the occurrence of a given label at a given position in the MRF over the whole chain. During this iterative search, occasional increases in the posterior energy of the MRF are allowed, which prevents getting trapped in a local energy minimum.

The most popular MCMC samplers are the Gibbs [Geman 84] and the Metropolis sampler [Metropolis 53]. The Metropolis sampler starts from some initial realization $\mathbf{x}$ of an MRF and at each step, a new candidate realization is obtained by random perturbation of the previous realization. Then the change in the posterior energy ΔE is computed and the new realization is accepted if $\Delta E \leq 0$ and accepted with probability p if $\Delta E > 0$. In practice, this means that a random number with uniform distribution on $[0, 1)$ is generated and compared with $\exp(-\Delta E/T)$, where T is the temperature in Gibbs distribution (Eq. (2.3)). One iteration of this algorithm is completed once all the labels are updated. The algorithm runs through multiple iterations until convergence,

which is achieved when the equilibrium or maximum number of iterations is reached.

The Gibbs sampler is the special case of the Metropolis sampler, where the next realization is based on the conditional probability rather than the energy change. Specifically, instead of generating a single n-dimensional vector in a single pass using a full distribution, the Gibbs sampler generates a sample for n random variables sequentially from n univariate conditional distributions (i.e., distributions in which all the random variables are fixed except one). Due to this, it is usually faster and easier to use than the Metropolis sampler. However, its application is limited to the problems where all the conditional distributions of the target distribution can be sampled exactly.

Note that the above mentioned MCMC samplers operate at a fixed temperature T. The quality of the solution can be further improved if this temperature, or some other parameter, is gradually reduced from very high value to a value close to zero in an iterative process, which helps substantially to avoid getting trapped in the local minimum (see Section 2.2.2). At each step of this process, i.e., at each value of T, a sampling method (e.g., Metropolis sampler) is applied. After the sampling method converges at the current value of T, T is decreased according to a carefully chosen schedule. This optimization method is called simulated annealing [Kirkpatrick 83, Cerny 85]. The most popular, and the only theoretically justified type of annealing, is stochastic simulated annealing [Geman 84], which employs the Metropolis or the Gibbs sampler at each value of T. Other annealing algorithms are listed in [Li 95], such as deterministic graduated non-convexity [Blake 87] and mean field annealing [Peterson 89]. The drawback of annealing methods is that they are extremely computationally intensive, because the temperature T has to be decreased gradually according to some schedule [Kirkpatrick 83, Geman 84], and at each value of T the sampling method has to converge.

2.3.2 Iterated conditional modes

Iterated conditional modes (ICM) is a simple, "greedy" inference method aiming at approximate MAP estimates. It starts from an initial estimate and then visits the nodes in some predefined order. While the true MAP estimate would maximize the posterior probability $P(\mathbf{x}|\mathbf{y})$, in the case of ICM in each iteration the new estimate $\hat{x}_i$ at node i maximizes the conditional probability given the evidence $\mathbf{y}$ *and* the current estimation $\hat{\mathbf{x}}_{S\setminus i}$ elsewhere:

$$\hat{x}_i = \arg\max_{x_i} P(x_i|\mathbf{y}, \hat{\mathbf{x}}_{S\setminus i}). \tag{2.16}$$

Due to the Bayes theorem (Eq. (2.9)), the assumption of conditional independence of observations (Eq. (2.11)) and the Markov property (Eq. (2.1)), it follows that the approximate posterior probability is [Besag 86, Pižurica 02a]

$$P(x_i|\mathbf{y}, \hat{\mathbf{x}}_{S\setminus i}) \propto P(y_i|x_i)P(x_i|\hat{\mathbf{x}}_{\partial i}). \tag{2.17}$$

Combining the above two equations, the ICM update rule becomes

$$\hat{x}_i = \arg\max_{x_i} P(y_i|x_i)P(x_i|\hat{\mathbf{x}}_{\partial i}). \tag{2.18}$$

If we look at the pairwise MRF, which is most often used, we can express the spatial interactions with the pairwise clique potential (see Section 2.2.2) as

$$P(x_i|\hat{\mathbf{x}}_{\partial i}) \propto \exp\Big(-\sum_{j\in\partial i} V_{ij}(x_i, \hat{x}_j)\Big). \tag{2.19}$$

Therefore, the ICM estimate becomes

$$\hat{x}_i = \arg\max_{x_i} P(y_i|x_i) \exp\Big(-\sum_{j\in\partial i} V_{ij}(x_i, \hat{x}_j)\Big). \tag{2.20}$$

Once all the nodes are visited, one iteration of the algorithm is finished. The procedure is repeated until convergence, which is guaranteed to occur and in practice is very fast compared to other inference methods. This stands for the sequential update scheme, where labels are updated as nodes are being visited. In the parallel case, all labels are updated at once at the end of each iteration, which can result in small oscillations rather than convergence. In both cases, only a local maximum of the posterior probability is reached and the results highly depend on the initial estimate. The initial estimate is usually set to the maximum likelihood estimate $\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} P(\mathbf{y}|\mathbf{x})$, although better options may exist.

2.3.3 Iterated conditional expectations

To overcome certain limitations of ICM, which will be discussed below, the iterated conditional expectations (ICE) algorithm was introduced in [Owen 89], *specifically* for the Ising MRF model. Initially, ICE was developed for image segmentation applications and later it was applied to image restoration [Zhang 93]. In both cases, it was shown that ICE outperforms ICM for very noisy images. Unlike ICM, which assigns labels after visiting the node (in the sequential case) or after each iteration (parallel case), ICE within one iteration updates only the approximate a posteriori probabilities of labels and actually assigns labels only after all iterations are completed. ICE is closely related to the mean field theory [Bilbro 88]. Despite its potential, this method is much less known than ICM in image processing community and usually neglected in recent papers.

In order to explain ICE, we will first derive the ICM update rule for the Ising MRF model, as in [Owen 89]. Specifically, the Ising model with binary labels $x_i \in \Lambda = \{0, 1\}$ is considered, thus the pairwise potential is $V(x_i, x_j) = 0$ if $x_i \neq x_j$, and $V(x_i, x_j) = -\gamma$ if $x_i = x_j$, where $\gamma > 0$. This means that, e.g., for the label $x_i = 1$ of the current node, the neighbourhood influence in ICM, i.e., the prior information from Eq. (2.19) is

$$P(x_i = 1|\hat{\mathbf{x}}_{\partial i}) \propto \exp(\gamma n_i), \tag{2.21}$$

where n_i is the number of nodes in the neighbourhood that are assigned the label 1 ($\hat{x}_j = 1$, where $j \in \partial i$), i.e., whose labels are estimated as 1. The neighbouring nodes that are assigned the label 0, $\hat{x}_j = 0$, do not have any influence since $V(x_i = 1, x_j = 0) = 0$. The same applies for the label $x_i = 0$, just that the spatial influence becomes the number of assigned (estimated) zeros in the neighbourhood ∂i. Therefore, for this type of model, ICM reduces the spatial-context information to the number of estimated labels of each type within the neighbourhood.[4] The approximate posterior probability in ICM is then

$$P(x_i = 1|y_i, \hat{\mathbf{x}}_{\partial i}) = \alpha P(y_i|x_i = 1) \exp(\gamma n_i), \tag{2.22}$$

where α is the normalization constant ensuring that all approximate posterior probabilities sum to one.

However, in [Owen 89] it was noted that this reduction of spatial-context information in ICM due to the immediate assignment of labels, introduces some loss of information. For example, the prior information for the current node with the label $x_i = 1$ is $P(x_i = 1|\hat{\mathbf{x}}_{\partial i})$. This probability has the same value, regardless of whether all neighbours $j \in \partial i$ of the node i have been assigned the label 1 ($\hat{x}_j = 1$) with probability $P(x_j = 1|y_j, \hat{\mathbf{x}}_{\partial j}) = 0.51$ or with probability 0.99. However, the probability $P(x_i = 1|\hat{\mathbf{x}}_{\partial i})$ would have a quite different value if, $\forall j \in \partial i$, $P(x_j = 1|y_j, \hat{\mathbf{x}}_{\partial j}) = 0.49$, which means that all the neighbours have been assigned the label 0 ($\hat{x}_j = 0$).

Therefore, ICE suggests to postpone label assignment until all iterations have finished, and update only the approximate a posteriori probabilities of the labels in each iteration. These a posteriori probabilities (e.g., for label 1) are now $P(x_i = l|y_i, \mathbf{x}_{\partial i})$, which is different from a posteriori probability $P(x_i = 1|y_i, \hat{\mathbf{x}}_{\partial i})$ in ICM (Eq. (2.22)), because labels in ICE are not estimated (assigned) in each iteration. For notational simplicity, we will from now on use the following abbreviations: $p_i(l) = P(x_i = l|y_i, \mathbf{x}_{\partial i})$ and $P(y_i|l) = P(y_i|x_i = l)$, where $l \in \{0, 1\}$. The ICE update rule for the Ising MRF for $p_i(1)$ is then defined as

$$p_i(1) = \alpha P(y_i|1) \exp\big(\gamma \sum_{j \in \partial i} p_j(1)\big) \tag{2.23}$$

$$\alpha = \Big(P(y_i|1) \exp\big(\gamma \sum_{j \in \partial i} p_j(1)\big) + P(y_i|0) \exp\big(\gamma \sum_{j \in \partial i} (1 - p_j(1))\big)\Big)^{-1}. \tag{2.24}$$

The posterior probability of the label $x_i = 0$ is then $p_i(0) = 1 - p_i(1)$. After a given stopping criterion (convergence or a pre-defined number of iterations), the labels are chosen so as to maximize these approximate posterior probabilities over two labels: $\hat{x}_i = \arg\max_{x_i} p_i(x_i)$. In practice instead, one can simply check if $p_i(1) > 0.5$ and assign $\hat{x}_i = 1$ if it is true and $\hat{x}_i = 0$ otherwise.

[4] This also stands for the Potts model [Potts 52] (see also Eq. (2.7)).

Comparing the approximate posterior probability of ICE from Eq. (2.23) with the one of ICM from Eq. (2.22), we can see that the n_i, i.e., the number of nodes assigned the label 1 in the neighbourhood ∂i, is replaced with $\sum_{j \in \partial i} p_j(1)$. This means that ICE considers all nodes j in the neighbourhood via the posterior $p_j(1)$. In this way, more information is kept in the inference process than it is the case for ICM.

In Section 2.4, we will propose a novel inference method, which also represents a generalization of ICE beyond the Ising MRF model, as we will show in Section 2.4.3.

2.3.4 Graph cut and its variations

Graph cut (GC) was first introduced in computer vision by [Greig 89], where it was demonstrated that it gives exact solution, i.e., global optimum, for a submodular PBF MRF problem (see Section 2.2.3 for the definition of a submodular PBF MRF). This algorithm was too slow for practical use on images, just like some other polynomial-time algorithms for minimizing submodular functions [Orlin 07]. However, it was shown in [Boykov 04] that for quadratic pseudo-Boolean functions (QPBFs), which are PBFs of at most two variables, the optimization can be performed much more efficiently. This is because the optimization problem in this special case can be reduced to the so-called $s-t$ min-cut problem on a graph, which is a classical combinatorial problem present in many applications and for which efficient algorithms have been developed [Ford 62, Dinic 70]. Boykov and Kolmogorov [Boykov 04] also proposed new optimization algorithms, which improve empirical performance of these standard techniques.

In order to minimize the energy from Eq. (2.14) with the $s-t$ min-cut, first the undirected graph of MRF nodes must be transformed into a directed weighted graph, which includes two additional terminal nodes, the source s and the sink t (see Fig. 2.4). These terminal nodes correspond to each of the binary labels, i.e., $s = 1$ and $t = 0$. The data term and the pairwise potential in Eq. (2.14) are represented by the weighted edges in the graph, with the restriction that the weights must be non-negative. An $s-t$ cut is defined as a set of edges that separate the source s from the sink t when removed. The cost of the cut is the sum of the weights of edges in the cut. The minimum solution $\hat{\mathbf{x}}$ is obtained by finding the $s-t$ cut with the minimum cost. This leaves the nodes connected either to s or t, meaning they are assigned with one of the two labels.

GC can also be used to solve the problems with multiple labels ($L > 2$). The most popular algorithms are the so-called *move-making* algorithms, which decompose the problem with multiple labels into a set of problems defined over binary labels, and this can be solved efficiently for certain types of problems, as explained above. Binary labels represent the decision either that the node keeps its old label or that it switches (moves) to the proposed label. The algorithm starts from some initial labelling and then iteratively finds the optimal subset of nodes (i.e., the one giving the largest energy

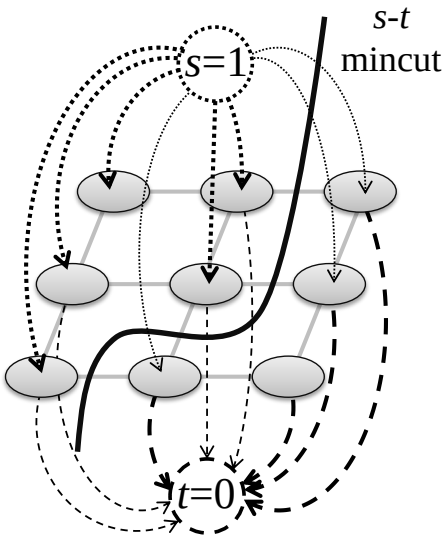


Figure 2.4: Example of a directed weighted graph with terminal nodes source s and sink t and an $s-t$ min-cut. Thicker arrows indicate the assignment of labels to regular nodes with one of the two labels corresponding to two terminal nodes.

decrease) to switch to the proposed labels, until energy cannot be further decreased by available moves. Available moves are determined based on the size of the move space, i.e., the number of possible changes that can be made to the current solution. In [Boykov 01b], two move-making algorithms were proposed: α-*expansion* and $\alpha\beta$-*swap*.

The power of the these algorithms is that substantial changes can be made to the current solution because multiple nodes can change their labels at the same time, while in ICM for example, only one node is allowed to do so. This enables the algorithm to avoid getting stuck in local minima, as well as faster convergence and independence of the initial labelling [Blake 11]. On the other hand, these algorithms are still approximate (for multiple labels) and their applicability is limited only to special kinds of problems (see [Boykov 01b] for more details). However, some attempts have been made to optimize a non-submodular energy function based on the roof duality relaxation of the integer programming problem [Rother 07], as well as to apply move-making algorithms to higher-order MRFs (e.g., in [Kohli 07, Kohli 09]).

2.3.5 Loopy belief propagation and its variations

Belief propagation (BP) falls into the category of *message-passing* algorithms. It was first introduced by Pearl in [Pearl 88], where it was shown that finding a global optimal solution is guaranteed, i.e., it is an exact inference al-

gorithm, but only in tree-structured graphs. There are two versions of BP: *max-product*, which produces joint MAP estimates, and *sum-product*, which leads to computation of marginals of individual random variables. Application of BP in graphs with loops, such as lattice graphs found in image processing and computer vision problems (see Fig. 2.1), was generally not advised, since the algorithm may not converge or it may give inaccurate results, as it was indeed shown for some examples in [Murphy 99]. However, its effectiveness was proved experimentally also in those graphs, especially in the decoding algorithm for error-correcting codes [Frey 98], known as "Turbo Codes" [Berrou 93], but also in some computer vision problems [Freeman 00, Blake 11]. Thus, BP applied to graphs with loops, being now an approximate inference algorithm, is called loopy belief propagation (LBP). We will focus in this thesis on the max-product version of the LBP algorithm, since we are considering MAP-MRF labelling problems. For more detailed overview of BP and LBP, see for example [Pearl 88, Frey 98, Yedidia 01a, Yedidia 05].

The central concept of the algorithm is the message defined with the message-update rule as

$$m_{ij}(x_j) = \alpha \max_{x_i} \{\psi_{ij}(x_i, x_j)\phi_i(x_i, y_i) \prod_{k \in \partial i:\ k \neq j} m_{ki}(x_i)\}, \tag{2.25}$$

(see Fig. 2.5(a) for graphical representation), where α is a normalization constant and the terms $\psi_{ij}(x_i, x_j)$ and $\phi_i(x_i, y_i)$ were previously introduced in Eq. (2.15). $\psi_{ij}(x_i, x_j)$ is called the pairwise *compatibility* and it represents the statistical dependency between pairs of labels of neighbouring nodes, x_i and x_j, i.e., it encodes the prior information via MRF. $\phi_i(x_i, y_i)$ is called the *local evidence* (or likelihood, see Section 2.2.4) and it models the relationship between the labels and observation (measurement).

From Eq. (2.25), we can see that the message $m_{ij}(x_j)$ that the node i sends to its neighbouring node j, depends on the pairwise interaction $\psi_{ij}(x_i, x_j)$ between them, the local evidence $\phi_i(x_i, y_i)$ of i, and all the messages that node i received from all its neighbouring nodes except j. These messages are computed at each node and are sent to all its neighbours. Hence, the message $m_{ij}(x_j)$ from node i to node j can be interpreted as an "opinion" of node i about assigning label x_j to node j. This opinion also contains the information from other neighbouring nodes via messages that were sent to node i through the factor $\prod_{k \in \partial i:\ k \neq j} m_{ki}(x_i)$. Message update is conducted iteratively until convergence. However, LBP is not guaranteed to converge.

The second important term in LBP is belief, which is computed for each node *after* convergence of messages as

$$b_i(x_i) = \alpha \phi_i(x_i, y_i) \prod_{j \in \partial i} m_{ji}(x_i), \tag{2.26}$$

(see Fig. 2.5(b)). This equation says that the value of node's belief depends on its local evidence and on the product of all incoming messages into the node.

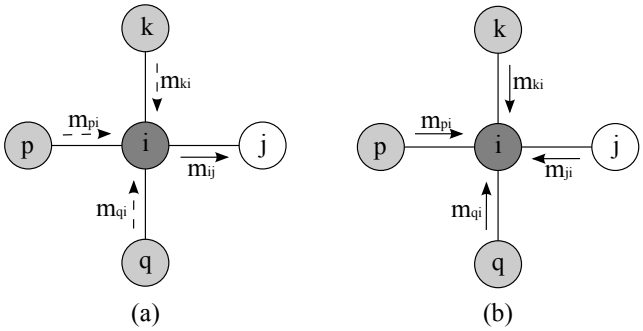


Figure 2.5: (a) Message-update rule - the message that node i sends to node j accumulates the messages that the node i has received previously from its neighbouring nodes, other than j. (b) Belief-update rule - belief of the node is calculated from *all* the incoming messages.

A belief actually approximates a posteriori probability of a node and it can be interpreted as a confidence of a node about its label. Therefore, in order to compute MAP estimates, at each node the label is chosen at the end of the algorithm so as to maximize the belief at that node:

$$\hat{x}_i = \arg\max_{x_i} b_i(x_i). \tag{2.27}$$

In recent years, various modifications of the original LBP algorithm appeared that attempt to correct some of its disadvantages. Firstly, it has been reported in [Yedidia 01b] that LBP has poor performance for graphs with many short loops and with both weak local evidence and strong compatibility constraints (meaning that the prior information has more influence on the result than observations). The poor performance is reflected in approximate beliefs being far from the exact ones and even MAP estimates being incorrect. Generalized belief propagation (GBP) [Yedidia 00, Yedidia 01b] has been proposed as a solution. Here, messages are exchanged between groups of nodes, because those are believed to be more informative. The second modification is the tree-reweighted max-product algorithm [Wainwright 05] that is guaranteed to produce correct MAP estimates under certain conditions. An interesting property of this algorithm is that it computes the lower bound on the energy from Eq. (2.13). However, the algorithm does not necessarily converge. There is also an improved version of the original algorithm, called sequential tree-reweighted message passing [Kolmogorov 06], where lower-bound estimate is guaranteed not to decrease, which results in certain convergence properties [Szeliski 08].

We investigated the use of LBP for detection of fine structures and thin edges using the Ising MRF model [Pižurica 02b], which will be explained in more detail in Section 2.5.2. The original version performed quite well with the Metropolis algorithm as inference engine. Therefore, we were expecting even better performance and possibility for further upgrade by using LBP.

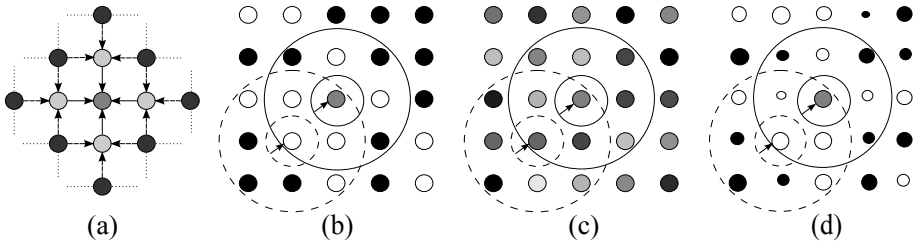


Figure 2.6: Graphical representation of information propagation through the binary MRF for different algorithms. The label of the central node is to be estimated. Grey node - label is to be chosen, black or white node - the label is set to one of the two values. (a) LBP: the messages propagate in pairwise fashion. (b) ICM: the decision is influenced by the estimated labels in the neighbourhood. (c) NCMP: the decision is influenced by the beliefs of neighbouring nodes. (d) WICM: the decision is influenced by the estimated labels in the neighbourhood and the confidences of their estimation. Different sizes of nodes represent different weights, i.e., node's confidence about its assigned label.

However, obtained results were unsatisfactory and even the experiments with GBP showed no improvement. This motivated us to develop a new inference algorithm, introduced in the next section, which performs well also in cases where LBP fails.

2.4 Neighbourhood-consensus message passing

In this section, we propose a novel inference method for MAP estimation with MRF priors. The central idea is to integrate a kind of joint "voting" of neighbouring labels into a message-passing scheme similar to LBP. While LBP operates with many pairwise interactions, we define "messages" sent from a neighbourhood as a whole. Hence the name neighbourhood-consensus message passing (NCMP). The practical algorithm is much simpler than LBP and combines the flexibility of ICM with some ideas of more general message passing. The proposed method can also be viewed as a generalization of ICE: we introduced ICE in Section 2.3.3 for the Ising MRF model and here we show that it can be interpreted as a particular instance of our general message-passing framework. We also develop a simplified version of NCMP, called weighted iterated conditional modes (WICM), that is suitable for large neighbourhoods.

2.4.1 Motivation and terminology

Although LBP and GC give state-of-the-art results in many computer vision and image processing applications [Szeliski 08], they still suffer from certain

disadvantages. GC is applicable only for certain class of problems (see Section 2.3.4) and the patch-based methods we investigate in this thesis do not belong to this class. Therefore, we concentrate more on LBP because of its generality. On the other hand, LBP has been reported to fail for graphs with huge number of nodes and many short loops [Yedidia 01b]. Our idea is to simplify LBP algorithm and make it better suited for this kind of graphs. In this situation, messages are unable to convey the necessary information globally throughout the graph. The solution can be to observe a larger neighbourhood, but then the speed, complexity and memory requirements become issues. We believe that these problems are caused by the message being defined as a pairwise interaction between two neighbouring nodes (see Fig. 2.6(a)). The number of these messages grows rapidly with the increase of the number of nodes in the graph and with the size of the neighbourhood. Our approach is to send one joint message from the whole neighbourhood to the central node rather than having a set of nodes that individually send messages to the central node like LBP does. In this way, we use some properties of ICM, which also consults the whole neighbourhood. The important difference is that the labels in ICM are estimated in each iteration (Fig. 2.6(b)), thus neglecting any confidence regarding their estimation (see the discussion in Section 2.3.3), while our approach takes this confidence into account through iterations and postpones the estimation of labels until the end of the algorithm. Therefore, we aim at finding a compromise between ICM and LBP, which is simpler and faster than LBP, while achieving better results than ICM and LBP.

To derive the new algorithm, we start from interpretations of the two basic terms in LBP, introduced previously in Section 2.3.5: message and belief. The *message* $m_{ij}(x_j)$ should express the opinion of the sending node i for each label of the receiving neighbouring node j. Remember that in LBP this opinion depends on the local evidence $\phi_i(x_i, y_i)$, the pairwise compatibility between the two nodes $\psi_{ij}(x_i, x_j)$ and the incoming messages from other nodes (Eq. (2.25)), Fig. 2.5(a)): $m_{ij}(x_j) = f(\phi_i(x_i, y_i), \psi_{ij}(x_i, x_j), m_{ki}(x_i))$, where $\forall k \in \partial i : \; k \neq j$. This message is computed for each edge, i.e., between each pair of nodes and in both directions between these nodes. Therefore, while computing messages, neighbourhood is not observed as one complete entity.

Our idea is to consult all neighbours of a node at once in order to make a decision about its label. ICM achieves this in a simple manner: just by counting the labels of each kind that are already estimated within the neighbourhood (Fig. 2.6(b)). If we look at ICM as a rather simple version of message passing, then the opinion of the neighbourhood for the labels of a central node consists of estimated labels of the nodes within that neighbourhood. It is hidden in the term $P(x_i|\hat{\mathbf{x}}_{\partial i}) \propto \exp\Big(-V_{i,\partial i}(x_i, \hat{\mathbf{x}}_{\partial i})\Big)$ (Eq. (2.18) and Eq. (2.20)), which can be viewed as a joint message that the neighbourhood sends to the central node. This way we could depart from pairwise interactions to potentially gain more freedom in defining spatial dependencies between the nodes. This formulation of ICM will be discussed in more detail in Section 2.4.4.

The second term in LBP is the node's *belief* $b_i(x_i)$, which can be

interpreted as the confidence of a node about its label. The belief depends on the local evidence and on the incoming messages into the node (Eq. (2.26), Fig. 2.5(b)): $b_i(x_i) = f(\phi_i(x_i, y_i), m_{ji}(x_i))$, where $\forall j \in \partial i$. For ICM, this belief is equal to one, since it greedily estimates the label at each iteration being completely confident about it. This is an obvious limitation. We wish to form the messages based on the "voting" of neighbouring labels and their beliefs, in the fashion of ICE. The ICE algorithm will be a particular instance in our general approach to *neighbourhood-consensus message passing* (NCMP).

2.4.2 NCMP framework

The main underlying idea of our work is to propagate belief through the graph by sending a single joint message to each node from its whole neighbourhood. We define a joint message from neighbourhood ∂i to node i as a function of the *neighbourhood potential* $V_{i,\partial i}(x_i, \mathbf{x}_{\partial i})$ and the *neighbourhood belief* $b_{\partial i}(\mathbf{x}_{\partial i})$ (Fig. 2.6(c)):

$$m_{\partial i \to i}(x_i) = f\Big(b_{\partial i}(\mathbf{x}_{\partial i}), V_{i,\partial i}(x_i, \mathbf{x}_{\partial i})\Big). \tag{2.28}$$

In particular, this function is defined as

$$m_{\partial i \to i}(x_i) = \exp\Big(- b_{\partial i}(\mathbf{x}_{\partial i}) V_{i,\partial i}(x_i, \mathbf{x}_{\partial i})\Big). \tag{2.29}$$

Note that if $\#\partial i = N$, where $\#$ denotes the cardinality of the set, and $x_j \in \{1, \ldots, L\}$, where $j \in \partial i$, there are L^N possible label combinations for $\mathbf{x}_{\partial i}$. Thus, $b_{\partial i}(\mathbf{x}_{\partial i})$ is a vector of L^N elements, $V_{i,\partial i}(x_i, \mathbf{x}_{\partial i})$ is a $L^N \times L$ matrix and the exponent in Eq. (2.29) contains their matrix multiplication. This is because we are considering all possible labels at each node and only assigning labels at the end of the algorithm. Therefore, all the label combinations in the neighbourhood ∂i of the current central node i participate in forming a *unified* opinion of the neighbourhood regarding the labelling of i, which is represented via the joint message $m_{\partial i \to i}(x_i)$. This opinion is based on the belief of label combinations within the neighbourhood and their spatial interaction with the central node, expressed through the neighbourhood potential. This joint message defined in a general form implies broad spectrum of possibilities in MRF modelling.

Further on, we define the neighbourhood belief as a function of node beliefs

$$b_{\partial i}(\mathbf{x}_{\partial i}) = f\Big(b_j(x_j)\Big), \tag{2.30}$$

where $j \in \partial i$. The *node belief* $b_i(x_i)$ is defined, in analogy to the classical belief definition from Eq. (2.26), as

$$b_i(x_i) = \alpha \phi_i(x_i, y_i) m_{\partial i \to i}(x_i), \tag{2.31}$$

where α is a normalization constant because beliefs have to sum up to one. Note that in this formulation, instead of separate messages from each neighbouring

node in Eq. (2.26), now one joint message per neighbourhood affects the value of belief. This gives us more flexibility in defining the spatial interactions between the nodes. Furthermore, unlike in LBP, where beliefs are computed only at the end of the algorithm, here we compute them in each iteration. The node belief represents an approximate a posteriori probability of node's label.

By looking at Eqs. (2.29), (2.30) and (2.31), we can see that our joint message contains similar components as the message in LBP (Eq. (2.25)). In particular, it explicitly depends on the neighbourhood potential, i.e., the prior information about the spatial context in the image, and implicitly, via the neighbourhood belief, it depends on the local evidence and the recursive messages received by the neighbouring nodes. However, as mentioned before, now this message expresses unified opinion of the neighbourhood, which limits pairwise interactions and allows richness in modelling compared to LBP.

We can define the neighbourhood potential $V_{i,\partial i}(x_i, \mathbf{x}_{\partial i})$ as a sum of clique potentials $V_C(\mathbf{x}_C)$ within a given neighbourhood, as explained previously in Section 2.2.2. The joint message in Eq. (2.28) then becomes

$$m_{\partial i \to i}(x_i) = \exp\Big(- \sum_{C \subset i \cup \partial i} b_{C\setminus i}(\mathbf{x}_{C\setminus i}) V_C(\mathbf{x}_C) \Big), \tag{2.32}$$

where $b_{C\setminus i}(\mathbf{x}_{C\setminus i})$ is the *clique belief* and $C\setminus i$ denotes a set of all nodes in the clique C except the central node i. Note that, as in the case of Eq. (2.29), the exponent contains matrix multiplication. In the case of pairwise cliques, the clique belief $b_{C\setminus i}(\mathbf{x}_{C\setminus i})$ from Eq. (2.32) reduces to the node belief defined in Eq. (2.31). In general, for cliques containing more than two nodes, we average the beliefs from Eq. (2.31) of the corresponding nodes

$$b_{C\setminus i}(\mathbf{x}_{C\setminus i}) = \beta \sum_{j \in C\setminus i} b_j(x_j), \tag{2.33}$$

where $\beta = \frac{1}{\#\{C\setminus i\}}$.

Like in ICM, we need to start from some initial configuration. In practice, we form the initial mask by maximum likelihood estimation, $\hat{x}_i = \arg\max_{x_i} \phi_i(x_i, y_i)$, and then we initialize belief of each node by setting it to the value that favours the label of that node in the initial mask. After initialization, the algorithm runs through iterations until some stopping criterion is satisfied or until the specified number of iterations is reached. We used the parallel-update scheme that calculates all the messages and beliefs based on the values from the previous iteration. At the end of the iterative algorithm, labels are assigned to nodes by maximizing their belief, $\hat{x}_i = \arg\max_{x_i} b_i(x_i)$. Since $b_i(x_i)$ approximates the a posteriori probability of x_i, this assignment solves our MAP-MRF labelling problem.

We introduced the proposed NCMP framework in a general form, which is not limited to pairwise interactions. This general form gives much room for improvement of the spatial prior, in terms of orientation selectivity (anisotropic models) and possibilities of using higher-order MRFs. MCMC

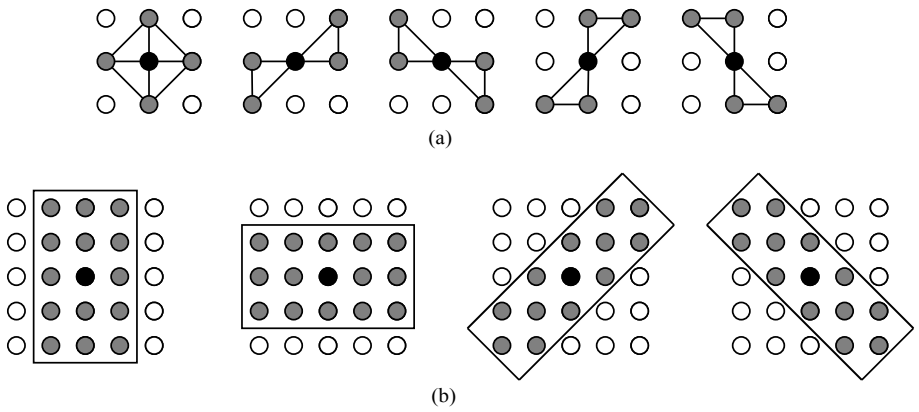


Figure 2.7: (a) Oriented sub-neighbourhoods from [Pižurica 02b]. (b) Another possible set of larger oriented neighbourhoods.

samplers, like the Metropolis sampler (Section 2.3.1), can perform inference in higher-order and anisotropic models. The Metropolis sampler was used to demonstrate the relevance of using, in particular, anisotropic models in [Pižurica 02b] via oriented sub-neighbourhoods depicted in Fig. 2.7(a). These oriented sub-neighbourhoods can better model edges in different directions and retain the homogeneity of the structure. We will illustrate the advantages of the anisotropic model in Section 2.5.2, and we will show that our NCMP method can handle this type of model, whereas standard LBP cannot.

2.4.3 NCMP as our generalization of ICE

Let us now focus on pairwise MRF, where all the cliques consist of two nodes $\langle i, j \rangle$. As mentioned in the previous subsection, in that case the clique belief $b_{C\setminus i}(\mathbf{x}_{C\setminus i})$ is equal to the node belief $b_j(x_j)$. From the joint message in Eq. (2.32) we can derive the following:

$$m_{\partial i \to i}(x_i) = \exp\Big(- \sum_{j \in \partial i} \sum_{x_j} b_j(x_j) V_{ji}(x_j, x_i) \Big). \tag{2.34}$$

Then the node belief from Eq. (2.31) becomes

$$b_i(x_i) = \alpha \phi_i(x_i, y_i) \exp\Big(- \sum_{j \in \partial i} \sum_{x_j} b_j(x_j) V_{ji}(x_j, x_i) \Big). \tag{2.35}$$

Remember that the node belief represents approximate a posteriori probability (Section 2.4.2), which we will denote by $p_i(x_i) = P(x_i | y_i, \mathbf{x}_{\partial i})$, and that the local evidence is equal to the likelihood $\phi_i(x_i, y_i) = P(y_i | x_i)$ (Section 2.2.4). Then we can rewrite Eq. (2.35) as

$$p_i(x_i) = \alpha P(y_i|x_i) \exp\Big(- \sum_{j \in \partial i} \sum_{x_j} p_j(x_j) V_{ji}(x_j, x_i) \Big). \tag{2.36}$$

Now, let us consider the Ising model from Section 2.3.3, which encourages assignment of equal labels to neighbouring nodes. In particular, the pairwise potential $V(x_i, x_j) = 0$ for $x_i \neq x_j$ and $V(x_i, x_j) = -\gamma$ for $x_i = x_j$, where $\gamma > 0$. We denote the pairwise potential as $V(x_i, x_j)$ because the Ising MRF model is homogeneous. Then from Eq. (2.36), and using abbreviations introduced in Section 2.3.3, it follows that the a posteriori probability of $x_i = 1$ is

$$\begin{aligned} p_i(1) &= \alpha P(y_i|1) \exp\Big(- \sum_{j \in \partial i} \big(p_j(0) V(0,1) + p_j(1) V(1,1) \big) \Big) \\ &= \alpha P(y_i|1) \exp\Big(- \sum_{j \in \partial i} p_j(1) V(1,1) \Big) \\ &= \alpha P(y_i|1) \exp\Big(\gamma \sum_{j \in \partial i} p_j(1) \Big). \end{aligned} \tag{2.37}$$

By comparing Eq. (2.37) with Eq. (2.23), we can see that they are identical. This means that ICE represents one particular instance of our NCMP framework for the Ising MRF model. However, our proposed NCMP framework is far more general than ICE, in the sense that it is applicable to more complex MRF models (see discussion at the end of Section 2.4.2), in addition to being a novel message-passing setting.

2.4.4 Weighted iterated conditional modes

In this subsection, we propose a simplified version of our NCMP framework, called weighted iterated conditional modes (WICM), where we assign labels to nodes at *each iteration*. Such assignment gives the proposed algorithm a discrete nature, in the fashion of ICM, rather than continuous one, as in ICE and our general NCMP formulation. However, unlike in ICM, we propagate additional information, which is the confidence of that assignment, by sending a joint message from the neighbourhood to the central node. This scheme is much simpler and can be of special interest when working with large neighbourhoods.

The assignment of a label to a node is conducted at each iteration by setting a label to the value that maximizes its node belief

$$\hat{x}_i = \arg\max_{x_i} b_i(x_i). \tag{2.38}$$

Node belief represents an approximation of a posteriori probability and is defined in Eq. (2.31). In words, node belief depends on the observation at the node, expressed via the local evidence $\phi_i(x_i, y_i)$, and the opinion of its *whole*

surrounding neighbourhood about each of its labels, expressed via the joint message $m_{\partial i \to i}(x_i)$ sent to it from the neighbourhood. However, this message has a different definition from the message in Eq. (2.32), because we assign labels at each iteration and we want only the probability of this assignment to influence the assignment of a label of the central node, rather than the probabilities of all possible label combinations within the neighbourhood. Therefore, the message in WICM is defined as

$$m_{\partial i \to i}(x_i) = \exp\Big(- \sum_{C \subset i \cup \partial i} b_{C \setminus i}(\hat{\mathbf{x}}_{C \setminus i}) V_C(x_i, \hat{\mathbf{x}}_{C \setminus i}) \Big). \tag{2.39}$$

We can then define the clique belief $b_{C \setminus i}(\hat{\mathbf{x}}_{C \setminus i})$, for example, as an average value of node beliefs that belong to the considered clique, as in Eq. (2.33). Note that both in the argument of the clique belief and the clique potential of Eq. (2.39), the labels of neighbouring nodes are already set to the value that maximizes a posteriori probability (via Eq. (2.38)), which is different from the message in a general NCMP framework (Eq. (2.32)). This means that the joint neighbourhood message depends on the specific (current) label configuration within that neighbourhood.

In order to motivate the name of the proposed algorithm, we shall first revisit ICM and place it within a unifying message-passing framework. If we look at Eq. (2.18) and compare it with Eq. (2.38) and Eq. (2.31), we can see that a message in ICM corresponds to the influence of the estimated labels in the neighbourhood, i.e.,

$$m_{\partial i \to i}(x_i) = P(x_i | \hat{\mathbf{x}}_{\partial i}), \tag{2.40}$$

which can be further developed via neighbourhood and clique potential into

$$\begin{aligned} m_{\partial i \to i}(x_i) &= \exp\Big(- V_{i,\partial i}(x_i, \hat{\mathbf{x}}_{\partial i}) \Big) \\ &= \exp\Big(- \sum_{C \subset i \cup \partial i} V_C(x_i, \hat{\mathbf{x}}_{C \setminus i}) \Big). \end{aligned} \tag{2.41}$$

Therefore, we can see that the message in WICM (Eq. (2.39)) is a weighted version of the message in ICM derived in Eq. (2.41) because we add weights to the clique potentials in the form of belief, while still estimating (assigning) labels in each iteration (see Fig. 2.6(d) for graphical representation). Hence the name weighted iterated conditional modes.

Due to sampling, i.e., choosing labels in each iteration, the proposed algorithm retains the simplicity of ICM, especially when it comes to generalization beyond pairwise potentials. On the other hand, it follows the idea of NCMP, in the sense that it is a message-passing algorithm where a single joint message is sent from the neighbourhood to the central node. However, a certain loss of information occurs in comparison with the general NCMP due to this sampling, making that way a trade-off between complexity and qualitative performance.

2.5 Experiments and results

In this section, we present a few example applications that illustrate the potential of the proposed NCMP approach. We consider both binary and multi-label MRFs with the second and first-order neighbourhoods. We compare the proposed approach with the reference methods, namely ICM, LBP and GC, on the same examples and the same set of parameters. However, note that LBP and GC cannot be used in all applications. In the binary denoising example of Section 2.5.1 and the binary segmentation example of Section 2.5.3, we used the code available at http://vision.middlebury.edu/MRF/ that accompanies comparative study of [Szeliski 08]. Apart from this, we used our own implementation of LBP and ICM in MatLab.

2.5.1 Noise removal from a binary image

In this example, the goal is to remove noise from an observed noisy image $\mathbf{y}$, whose pixel values are $y_i \in \{-1, 1\}$, $\forall i \in S$. We assume that the image is obtained by randomly flipping the sign of certain fraction of pixels in a noise-free image $\mathbf{x}$, $x_i \in \{-1, 1\}$, $\forall i \in S$. The likelihood, i.e., the relationship between the observation and the label, is $\phi(x_i, y_i) = \exp(\eta x_i y_i)$, where η is a positive constant. The spatial interaction of labels that favours clustering of the labels of the same type is modelled with the smoothness prior (Section 2.2.3) as $\psi(x_i, x_j) = \exp(-V(x_i, x_j))$, where $V(x_i, x_j) = -\gamma x_i x_j$ and γ is a positive constant. We used the second-order neighbourhood. This is a very simple example of MRF application but is typical of more sophisticated applications.

The performance of the algorithm is illustrated on a binary image from [Bishop 06]. The noisy version is obtained by randomly flipping 10% of the pixels in a noise-free image. The denoising result depends on the value of the parameters of the model η and γ. One way to estimate these parameters involves computing the partition function of the associated MRF, which is slow. Therefore, we determined the optimal parameters experimentally. The only significant noise reduction is obtained for $\eta = 0.5$ and $\gamma = 1.0$, the second-order neighbourhood and a maximum of 500 iterations, shown in Fig. 2.8. It is obvious that ICM gives by far the worst result because it quickly gets trapped in the local optimum. The proposed methods NCMP and WICM cope well with this type of problem and give comparable results with the state-of-the-art methods GC and LBP. The GC method gives the optimal solution for the energy function on a binary MRF in only one iteration, which makes it the fastest method, but some errors in pixel-labelling are noticeable, e.g., the letter "e" in the second row. Finally, if we compare the two proposed methods, WICM yields slightly poorer results than NCMP, because of isolated dots in the background, while still outperforming ICM.

We can also measure the quantitative performance by comparing the percentage of misclassified pixels with the original. For the above mentioned parameters corresponding to the results from Fig. 2.8, the percentages of misclassified pixels for different methods are shown in Table 2.1. The GC method

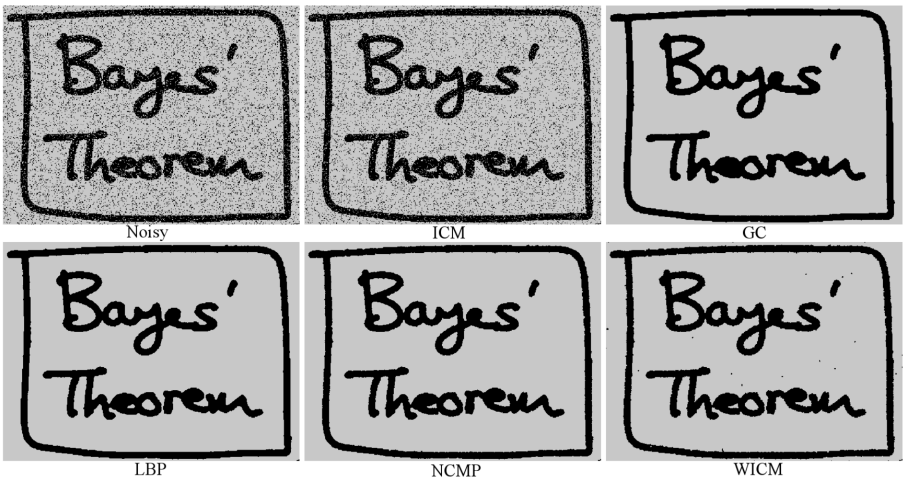


Figure 2.8: Top: noisy version of the image from [Bishop 06] of size 1259x1703 with 10% of the pixels flipped, and resulting images of ICM and GC. Bottom: resulting images obtained with LBP, NCMP and WICM.

Table 2.1: Comparison of misclassified pixel percentage for the results from Fig. 2.8 with parameters $\eta = 0.5$ and $\gamma = 1$ and 10% of the pixels flipped.

ICM	GC	LBP	NCMP	WICM
6.30	0.34	0.43	0.37	0.42

yields the smallest percentage of misclassified pixels, although the information content (text) result is in this case recovered more faithfully by LBP and NCMP (see Fig. 2.8). Other parameter values give significantly poorer results for all methods.

We also noted that the performance of all the methods depends on the size of the image being processed. The image in Fig. 2.8 is large in size, 1259x1703 pixels, so we also tested the performance on smaller images (80% and 50% of the original size). The results are summarized in Table 2.2 for parameters $\eta = 1$ and $\gamma = 0.5$ and also 10% of the pixels flipped. In general, for smaller images all the methods perform better and the difference in results of different methods becomes smaller. However, we can see that even in this case, our method outperforms ICM and LBP. GC gives the best quantitative result, but remember that it cannot be used in all applications, as discussed previously in Section 2.3.4.

In conclusion, in terms of quantitative comparison, both our methods perform better than LBP and ICM in all analysed cases, while being outperformed by the GC method. However, our method yields better qualitative result than GC, as illustrated in Fig. 2.8. In terms of speed, our method is

Table 2.2: Comparison of misclassified pixel percentage for different sizes of the test image from Fig. 2.8 with parameters $\eta = 1$ and $\gamma = 0.5$ and 10% of the pixels flipped.

% of the original size	ICM	GC	LBP	NCMP	WICM
100%	7.68	1.21	7.64	6.34	6.33
80%	0.36	0.07	0.23	0.14	0.14
50%	0.57	0.11	0.4	0.2	0.2

slower than GC because it requires multiple iterations, while GC reaches the optimum in one iteration for a binary MRF. However, our method is faster than LBP by an order of magnitude (for the same number of iterations, see later Fig. 2.15), in addition to being simpler for implementation.

2.5.2 Detection of signal of interest in wavelet domain

Lately, inferring the spatial structure in sparse image representations has become an important issue in structured-sparsity approaches [He 09, Huang 09, Baraniuk 10, Cevher 10, Pižurica 11]. In this example, we make an attempt to infer spatial clustering of sparse image coefficients, particularly by detecting signal of interest, i.e., meaningful edge coefficients, in noisy wavelet sub-bands. As illustrated in Fig. 2.9, true edge coefficients cannot be detected by simply thresholding the noisy sub-band.

One solution is to encode prior knowledge about spatial clustering of edge coefficients using an MRF model [Malfait 97, Pižurica 02b]. In this case, the labels of nodes $x_i \in \{-1, 1\}$ represent absence and presence of signal of interest, respectively. Conditional likelihoods describe distributions of the magnitudes of wavelet coefficients given each label, and we estimate these as described in [Pižurica 02b]. Spatial information is given by the isotropic model with pairwise potentials $V(x_i, x_j) = -\gamma x_i x_j$ (γ is a positive constant), that assigns a higher probability to edge continuity. This is actually the smoothness prior from Section 2.2.3 corresponding to the Ising model, but we used the second-order neighbourhood. By performing inference on this model, an edge map is obtained for each wavelet band that can be later used for subsequent processing in wavelet domain, e.g., for denoising.

The performance of the inference algorithms LBP, ICM, NCMP and WICM on the noisy wavelet sub-band from Fig. 2.9 is illustrated in Fig. 2.10. The result was obtained for a maximum of 20 iterations and $\gamma = 0.7$. Unlike in the previous application from Section 2.5.1, we can see that here LBP performs poorly because it deletes most of the edges leaving the mask to be barely recognizable. On the other hand, ICM gives quite good results in this example, better than LBP. Both NCMP methods perform slightly better than ICM, yielding more consistent edges, with clearer boundaries and without interruptions. We also included the results of the Metropolis sampler [Metropolis 53] (Section 2.3.1) that was used in [Pižurica 02b]. The advantage of the Metropo-

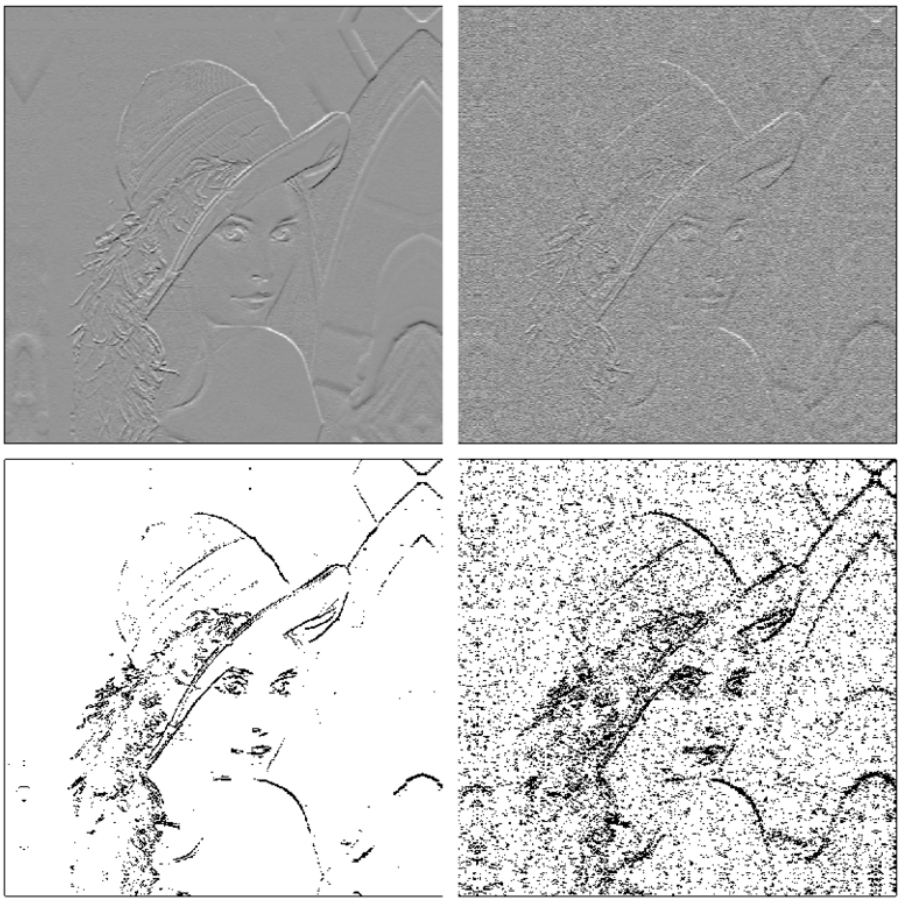


Figure 2.9: Top: noise-free and the corresponding noisy wavelet sub-band ($\sigma = 20$) of the "Lena" image. Bottom: detected edges by thresholding the two sub-bands, respectively.

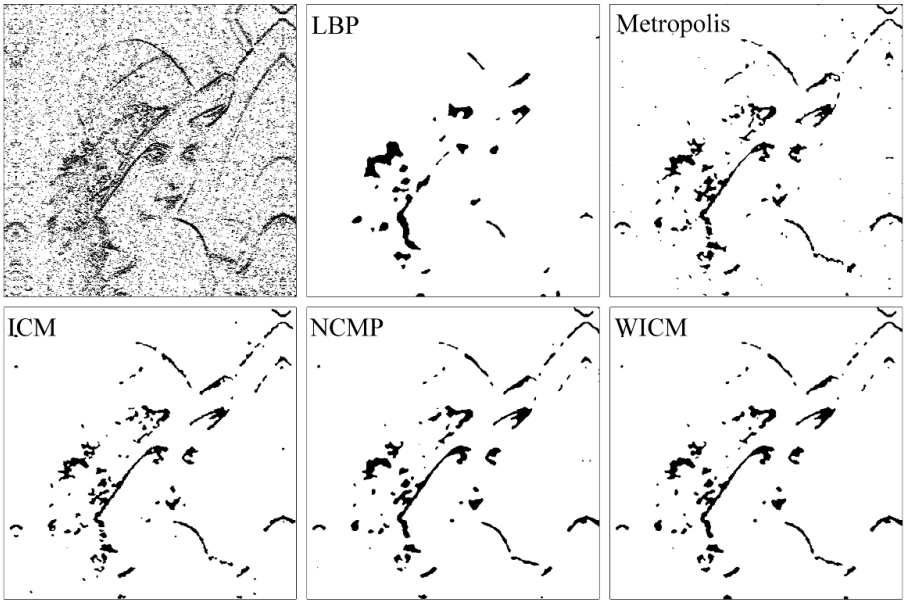


Figure 2.10: Masks of horizontal edges at the first scale of wavelet decomposition for $\sigma = 20$ obtained by using an *isotropic* MRF model. Top: initial mask and results of LBP and the Metropolis sampler. Bottom: results of ICM, NCMP, and WICM.

lis sampler is that it estimates accurately the a posteriori probabilities of each label, but in terms of the final binary mask, its performance in this example is comparable to ICM. Finally, we tested GC on this model, but due to the way the model is defined, it gave no meaningful results, i.e., it recognized no signal of interest in the noisy wavelet sub-band. Note that in this example the isotropic MRF model was used.

Another advantage of the proposed methods and ICM in comparison with LBP, is that they can be directly applied to anisotropic models, like those in Fig. 2.7. These models can be defined in such a way to further improve the results of edge detection. For example, in [Pižurica 02b], the potential of the sub-neighbourhood p is defined as $V_p(x_i, \mathbf{x}_{\partial i,p}) = -\gamma x_i \sum_{j\in\partial i,p} x_j$ and the potential of the complete second-order neighbourhood is $V_{i,\partial i}(x_i, \mathbf{x}_{\partial i}) = -\gamma x_i \max_p(\sum_{j\in\partial i,p} x_j)$. This choice of the neighbourhood potential results from the following reasoning: label $x_i = 1$, i.e., the presence of a signal of interest, should be assigned the high probability if any of the sub-neighbourhoods indicate the existence of a signal of interest, while label $x_i = -1$ is given preference if none of the sub-neighbourhoods has that indication.

The benefits of using the anisotropic model were already demonstrated in [Pižurica 02b] using the Metropolis sampler. In Fig. 2.11, we show that the performance of the proposed NCMP method also improves largely

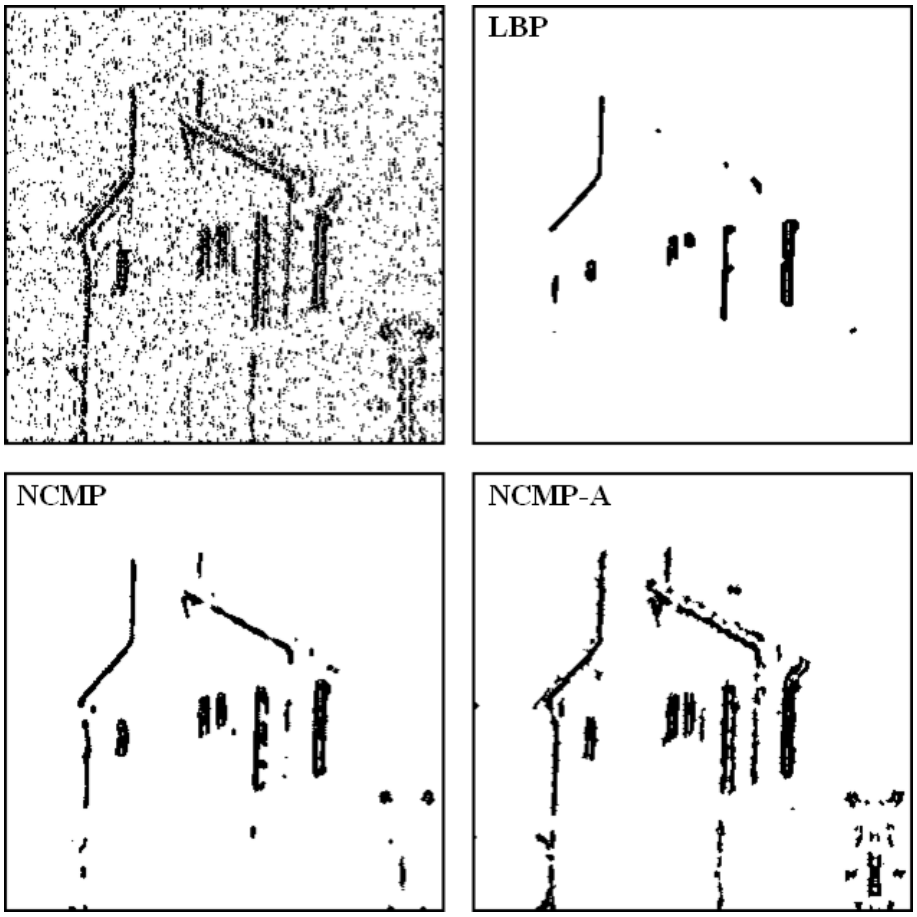


Figure 2.11: Masks of vertical edges at the first scale of wavelet decomposition for $\sigma = 30$. Top: initial mask and results of LBP for the isotropic model. Bottom: results of NCMP for the isotropic model and NCMP for the anisotropic model (denoted with an extension -A).

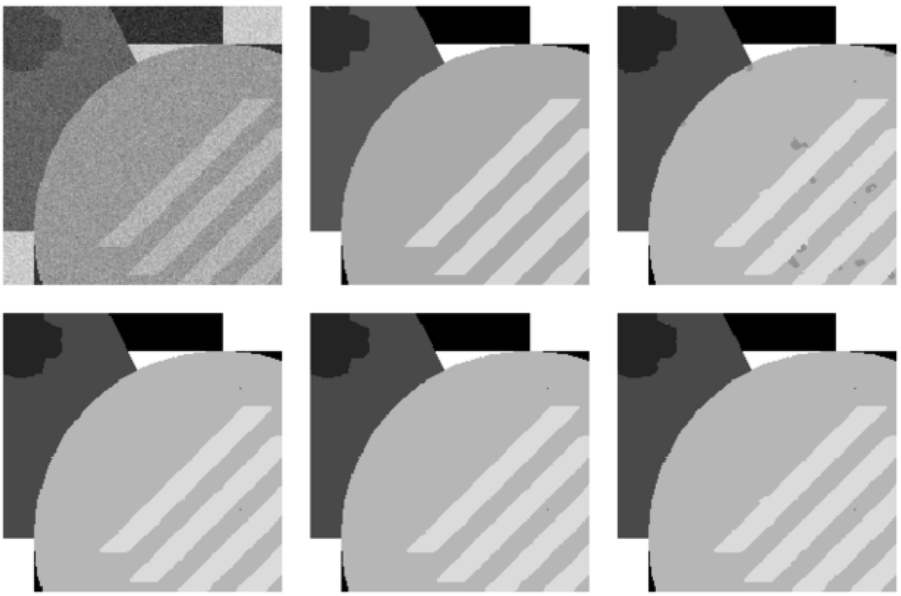


Figure 2.12: Cropped version of a synthetic image. Top: noisy image, ground truth and result of ICM. Bottom: results of LBP, NCMP and WICM.

with this anisotropic model (denoted as NCMP-A), in the sense that more edges are detected and structure is better preserved. NCMP gives similar results to that of the Metropolis sampler and ICM (not shown here). LBP yields worse results than NCMP for isotropic model also on this image. Additionally, it cannot be applied directly on the anisotropic model introduced above, and its performance is thereby inferior in this case.

2.5.3 Image segmentation

Another application is segmentation of a noisy image: each pixel is assigned one label that represents the segment to which the pixel belongs. In the first experiment, we used a synthetic image with artificially added white zero-mean Gaussian noise of standard deviation σ. The local evidence is then the Gaussian function with the mean value equal to the pixel value in the non-degraded image, which is here the label x_i, and the standard deviation σ:

$$\phi(x_i, y_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(\frac{-(y_i - x_i)^2}{2\sigma^2}\Big). \tag{2.42}$$

The pairwise potential is determined by the discontinuity preserving Potts model [Potts 52] (Eq. (2.7)).

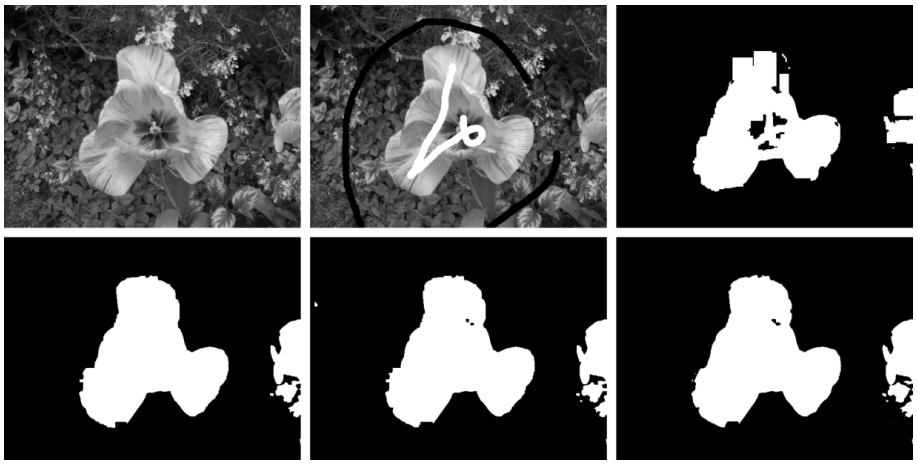


Figure 2.13: Binary segmentation of the "flower" image. Top (from left to right): original image, user data and result of ICM. Bottom (from left to right): result of GC, LBP and NCMP.

The segmentation results for $\sigma = 20$, the second-order neighbourhood, $\gamma = 1$ and maximum of 50 iterations are shown in Fig. 2.12. ICM has the weakest performance leaving obviously misclassified areas. NCMP and WICM yield similar results to LBP. GC (not shown) gave similar results to LBP and NCMP for the same setting.

In the second experiment, we apply the tested methods to the foreground/background segmentation from [Rother 04], where the test image from Fig. 2.13 is used and an MRF with the first-order neighbourhood. This method uses input from a user, who marked the foreground and background region (see middle image in the top row of Fig. 2.13). Again, our method outperforms ICM and yields a similar result to LBP. The GC method gives the best result for this example but note that its applicability is limited to special potential functions (see Section 2.3.4), while the proposed method is more general.

2.5.4 Super-resolution

For the super-resolution (SR) example, we used our approach from [Ružić 11b], where the MRF model was similar to [Freeman 00]. The details of this approach will be described later in this thesis in Section 3.4, but here we introduce it briefly for clarity. The idea is to find for each position in the unknown high-resolution (HR) image a well-matching patch from some candidate set of HR patches, so that it agrees well with the neighbouring overlapping patches and with the corresponding low-resolution (LR) content. The candidate set of HR patches is obtained from the previously formed database of pairs of LR/HR patches by the following procedure. First, the L most similar patches of each

input LR patch are found in the set of LR patches in the database, and then, their corresponding HR patches are taken as candidate patches. The local evidence is taken to be the matching error, i.e., sum of squared differences, between the starting LR patch and each of the found L most similar patches (see Eq. (3.15) for details). The pairwise potential is the error in the region of overlap between two neighbouring HR patches in the first-order neighbourhood (see Eq. (3.16)).

This problem is non-submodular [Rother 07] (see Eq. (2.6) for the definition of submodularity), which makes it difficult for GC. In [Rother 07], a simplified binary form of this problem was used for comparison of different inference algorithms with the proposed modification of GC. Here, we keep the original problem and compare the proposed method with LBP. Fig. 2.14 shows the cropped version of the "zebras" image. In the top row, on the left is the result of the standard bicubic interpolation algorithm (see Section 3.2.1), while on the right is the result of choosing the best match at each position, i.e., when no MRF modelling is applied. We can see that this approach produces sharper edges compared to the bicubic interpolation, which is an important aspect in SR, as will be discussed in detail in Chapter 3. However, it also introduces a lot of artefacts. In the bottom row are the results of LBP (left) and the proposed method (right). We can see that MRF modelling brings improvement, in the sense that artefacts are removed while the sharpness of edges is retained. Our method performs equally well as LBP, while being about ten times faster (see the right plot in Fig. 2.16, the relative performance for this application is similar).

2.6 Convergence consideration

In this section, we inspect the convergence properties of the proposed methods in comparison with LBP, ICM and GC for some of the application examples. In particular, we study the change of the total energy $E(\mathbf{x}, \mathbf{y}) = -\log P(\mathbf{x}, \mathbf{y})$ over iterations and/or time. The convergence plot for the application from Section 2.5.1 is shown in Fig. 2.15. On the left, we can see that ICM reaches the stable minimum already after the first iteration. However, this is the local energy minimum, even increased in comparison with the starting energy and much higher than that of the other methods. On the right of the same figure, we can see the zoomed-in plot with comparison of other methods (all except ICM). GC and WICM reach a stable minimum after the first and the third iteration, respectively, while LBP and NCMP oscillate. Both proposed methods reach the lower minimum than LBP and GC. Note that each of these iterations consists of 50 "inner" iterations, i.e., the total number of iterations is 500. The number of "inner" iterations, according to the code of [Szeliski 08], is the number of iterations which are conducted anyway, i.e., during which the value of the total energy of the MRF and the convergence of the algorithm are not checked.

In the example of segmenting a noisy image, we ran the algorithm for a maximum of 100 iterations and the energy of the last 90 iterations is

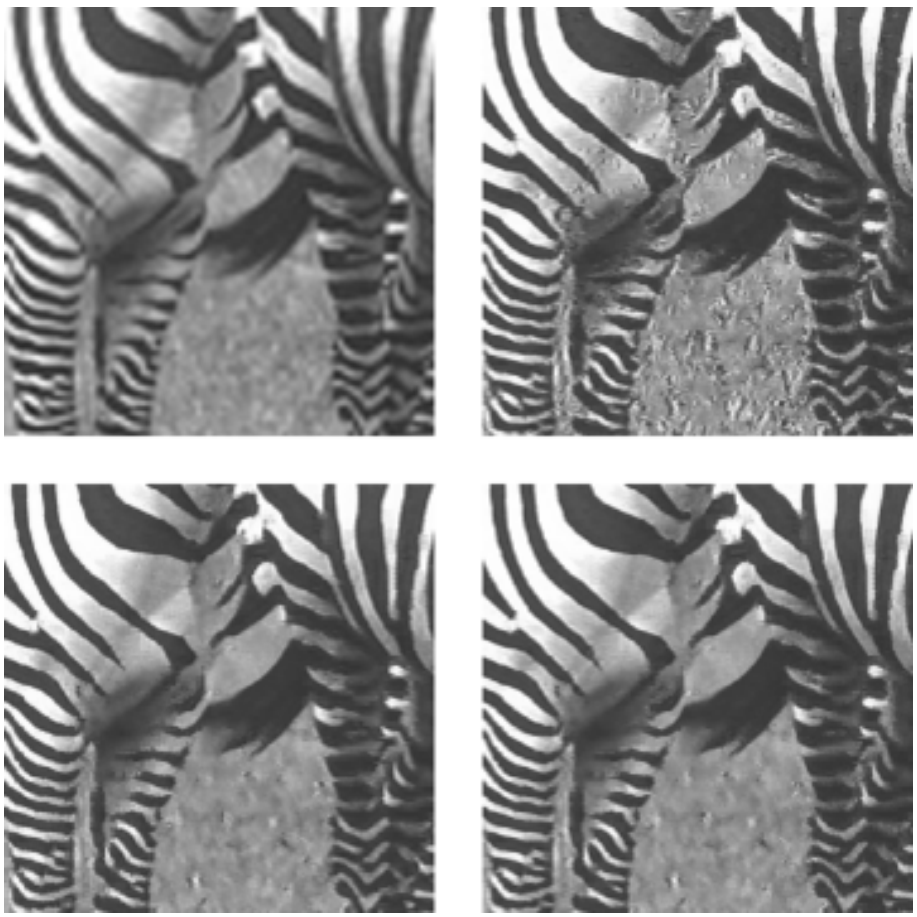


Figure 2.14: Cropped version of the "zebras" image with 2x magnification. From left to right and top to bottom: result of bicubic interpolation, best-match result, MRF result with LBP as inference method, and MRF result with NCMP as inference method.

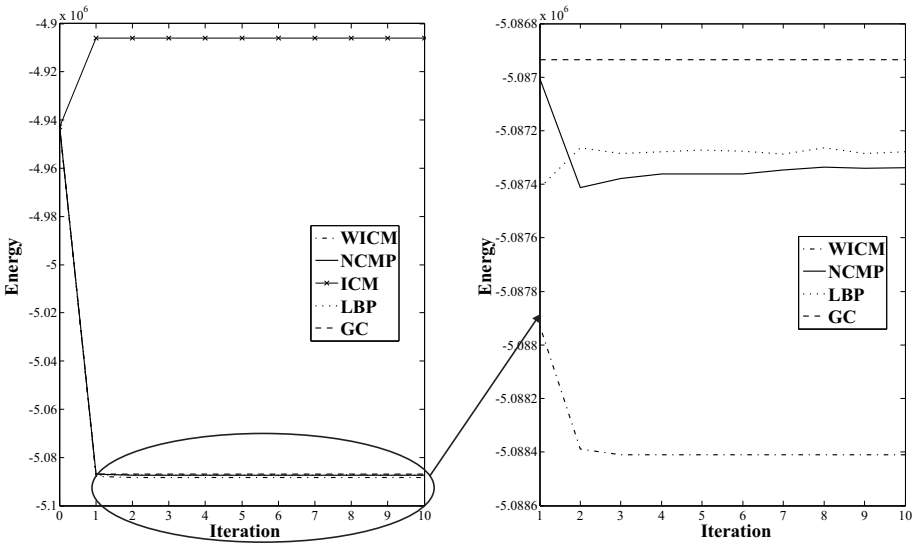


Figure 2.15: Illustration of the energy change over iterations for different algorithms on the example of noise removal from a binary image. Left: all algorithms. Right: zoomed-in part showing differences between WICM, NCMP, LBP and GC.

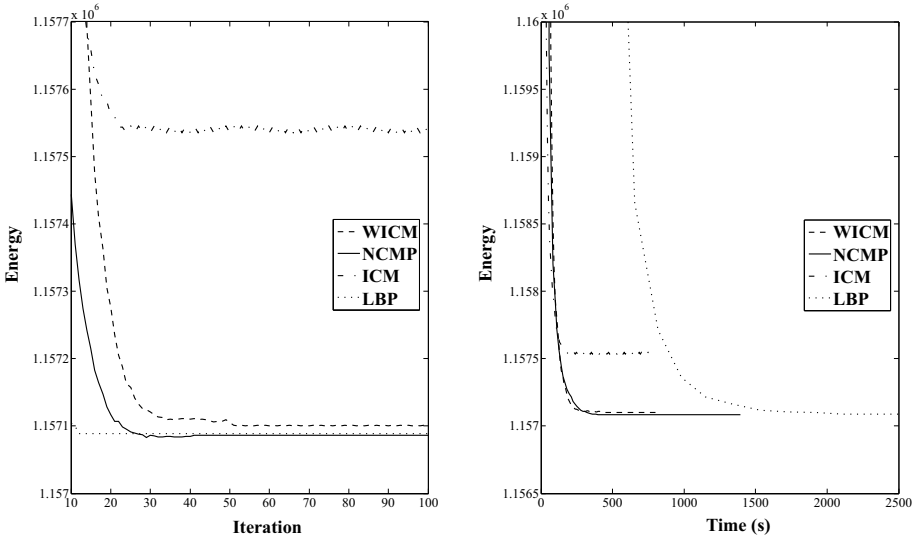


Figure 2.16: Illustration of energy change through iterations (left) and time (right) of different algorithms for the example of image segmentation.

shown on the left of Fig. 2.16. ICM has the highest energy minimum and it oscillates, as expected due to the parallel-update scheme. The other algorithms converge to the similar value. LBP converges in the least number of iterations (12), and it is the only steady one, in the sense that it is constantly decreasing the energy. WICM takes the highest number of iterations to reach the energy minimum, but then it oscillates, meaning that it follows the behaviour of ICM. NCMP has the lowest energy minimum, but only slightly different than LBP. It has some problems with stabilization of the energy (it increases and decreases) until eventually it converges. Although LBP converged in fewer iterations, it was still around five times slower than NCMP, as we can see on the right of Fig. 2.16, where the energy is plotted as a function of time.

2.7 Conclusion

In this chapter, we first introduced the MRF theory by explaining relevant definitions, Markov-Gibbs equivalence, examples of MRF models and the MAP-MRF labelling problem. Then we discussed in more detail inference methods, like MCMC sampler and simulated annealing, ICM and its variant ICE, GC and LBP.

The main contribution of this chapter is the new inference method based on message passing. We called this method neighbourhood-consensus message passing (NCMP) since a joint message is sent from the specified neighbourhood to the central node, which enables information to propagate through the graph. Information consists of beliefs of neighbouring nodes as confidence measure of their own labels. The proposed scheme has the computational simplicity and robustness similar to ICM, and the flexibility of a more general message passing. On the one hand, the performance is significantly improved in comparison with ICM, and on the other hand, we are working with a whole neighbourhood at once instead of pairs of nodes in comparison with LBP. Furthermore, we showed that the proposed method can be considered as a generalization of ICE for more complex MRF models. Additionally, we developed a simplified version of NCMP, called weighted iterated conditional modes (WICM), to overcome potential difficulties while working with larger neighbourhoods. Results on different example applications showed that the proposed methods outperform ICM, while giving comparable or, in some cases, favourable results in comparison with LBP in much shorter time. Finally, another contribution of this chapter is that we described some of the inference methods, namely ICM, ICE and LBP, within a unifying message-passing framework.

It is important to notice that the proposed NCMP is generally applicable to a wide range of problems, which allows us to employ it as an inference engine for patch-based methods, such as SR and inpainting. Furthermore, these patch-based methods are quite computationally intensive, so having a fast and simple inference method, such as NCMP, is of great benefit. The application of the proposed method to patch-based problems will be discussed in the following

chapters.

This work has led to a journal publication [Ružić 12c], as well as other two conference publications [Ružić 09, Ružić 11c], where the publication in [Ružić 09] was awarded with the Best Poster Award.

3

Patch-based image upscaling

Image upscaling plays an important role in image processing applications nowadays due to the huge amount of low-resolution (LR) video and image material that needs to be reproduced on high-resolution (HR) screens or printers. The low resolution is a consequence of using low-quality imaging sensors for image/video acquisition devices, such as webcams, cell phones and surveillance cameras. The task of image upscaling is to create an HR image from one or more LR images.

In this chapter, we will first introduce the problem of image upscaling in Section 3.1, together with its main applications. We formulate the image upscaling more formally and explain it within a regularization framework. In Section 3.2, we classify and give an overview of representative image upscaling methods. Among these methods, example-based (or patch-based) and multi-frame methods are considered to be super-resolution (SR) methods, since they are able to recover missing high frequencies in the HR image. Patch-based methods have become increasingly popular over the last decade for their ability to overcome the limitations of the multi-frame approach, above all the value of the magnification factor. Along this line, we developed a new patch-based method, presented in Section 3.4, that uses the input LR image itself as a search space for HR patches by exploiting image self-similarity across different resolution scales. The found HR patches are combined in an HR image by the means of Markov random field (MRF) models, where MRF is used to encode prior knowledge about the consistency of neighbouring HR patches. We employ our inference technique proposed in Chapter 2 to find the maximum a posteriori (MAP) estimate of the HR image. To obtain the final HR image, we apply back-projection and steering kernel regression as post-processing techniques. In this way, we are able to produce sharp and artefact-free results that are comparable or better than standard interpolation and state-of-the-art image upscaling techniques.

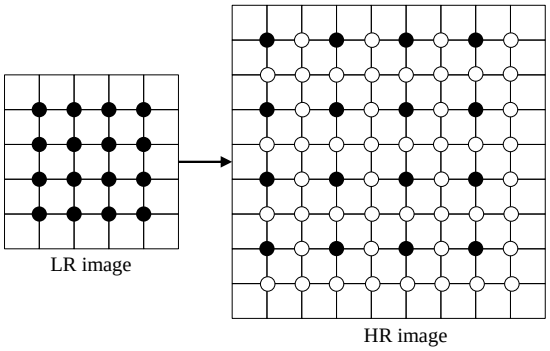


Figure 3.1: Upscaling procedure: LR pixels (black circles) are copied to the HR grid and the missing HR pixels (white circles) are to be estimated.

3.1 Introduction

Image upscaling refers to the problem of obtaining an HR image from one or multiple LR images by increasing their *pixel* resolution. Pixel resolution specifies image size, described by the total number of pixels in the image (e.g., expressed in megapixels), the number of pixels per each dimension (width $\times$ height) or the number of pixels per unit length or per unit area, such as pixels per inch (PPI) or pixels per square inch. Image upscaling can also be viewed as a re-sampling problem: an LR image constitutes a collection of discrete samples (pixels), obtained by sampling the continuous function of the scene at a certain (low) sampling rate. The goal is then to obtain an HR image by estimating the missing pixels (see Fig. 3.1).

Next to pixel resolution, there are also other ways to describe image resolution, e.g., spatial, spectral, radiometric. Especially important is *spatial* resolution, which is defined as the number of *independent* pixel values per unit length and as such, is related to the ability of distinguishing image details. For example, two images of the same size, i.e., of the same pixel resolution, can contain different amount of details, i.e., different spatial resolution. Increasing spatial resolution, i.e., increasing the level of detail in the image, inherently leads to image upscaling. Some image upscaling methods are capable of increasing spatial resolution. We will refer to those methods as *super-resolution* (SR), because the word "super" represents the ability of the technique to overcome the inherent resolution limitation of the LR imaging system by recovering the lost or degraded high frequencies during the acquisition process.

Spatial resolution can also be increased in a sensor by improving manufacturing techniques in order to reduce the pixel size or increase the chip size. This approach suffers from sensitivity to shot noise because the smaller the pixel size, the smaller amount of light available. With increasing spatial resolution, it is not possible to keep pixel size large enough: larger chips have higher capacitance and, therefore, slow charge transfer rate, making this ap-

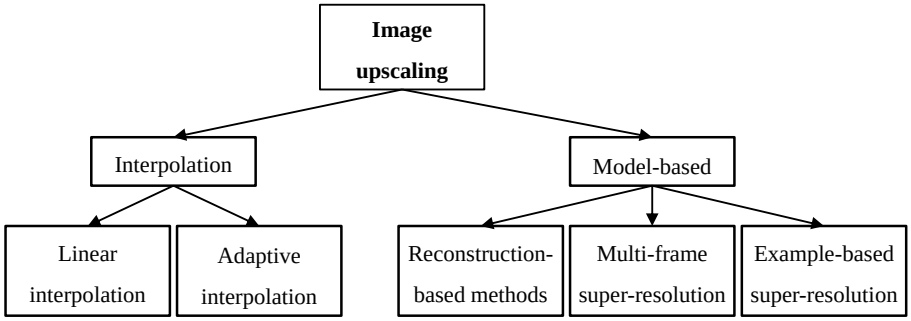


Figure 3.2: Graph representing different image upscaling methods.

proach ineffective [Park 03]. Therefore, image processing techniques for spatial resolution enhancement are needed. Furthermore, they allow the usage of existing LR imaging systems and LR image and video material.

We divide image upscaling methods into interpolation and model-based methods (Fig. 3.2). SR methods belong to the group of model-based methods, which assume the observation models described in Section 3.1.2 and treat image upscaling as an inverse problem, while assuming the image is corrupted with noise, blur and aliasing artefacts. Interpolation methods, on the other hand, assume that the continuous scene is sampled with Dirac pulses. All the approaches, except multi-frame SR, deal with image upscaling from a *single* input LR image. More in-depth overview of each group of methods will be given later in Section 3.2.

In this introductory section, we will review some of the applications of image upscaling and introduce observation models that relate LR and HR images. Furthermore, we will describe image upscaling in a regularization framework, since some image upscaling methods listed in Fig. 3.2 rely on these theoretical concepts.

3.1.1 Applications of image upscaling

Image upscaling plays an important role in image processing applications nowadays due to the huge amount of LR video and image material. Due to manufacturing limitations and cost restrictions, a lot of imaging material is still acquired in low resolution. Furthermore, a lot of LR material was captured with old equipment or according to some old standards, like NTSC and PAL recordings. Nowadays, this material must also be displayed on HR displays (e.g., high-definition television (HDTV)) or printed on HR printing devices (e.g., in document imaging) without visual artefacts. Therefore, an important application of image upscaling is to perform the conversion of this existing material from low to high resolution.

Image upscaling is already supported in digital cameras, e.g., the option *digital zoom* or in the demosaicing stage (reconstruction of full-colour

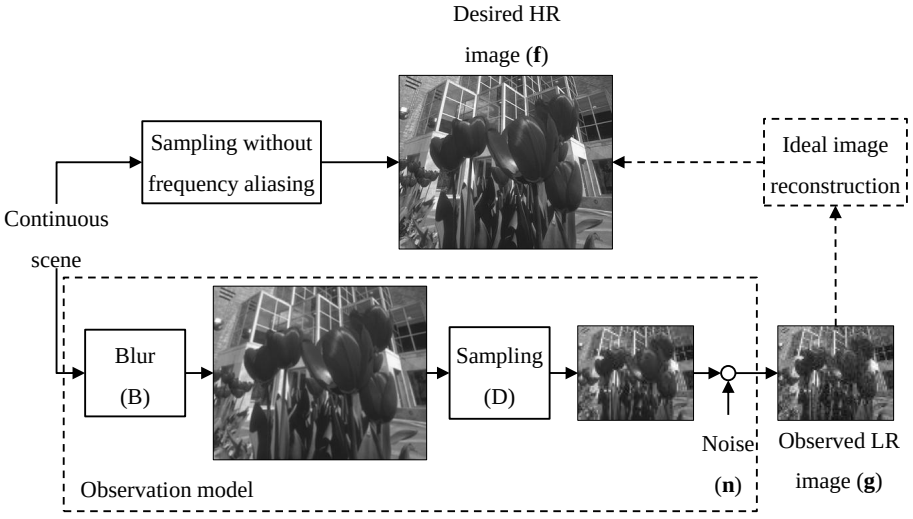


Figure 3.3: Observation model relating HR image with *one* observed LR image.

images from the colour-filtered CCD or CMOS samples), and in the image editing software, such as Adobe Photoshop, GIMP, Irfanview, etc.

Another important application is the zooming of region of interest in surveillance, forensics, satellite and medical imaging in order to facilitate content analysis. In surveillance or forensics, it is often necessary to recognize the face of a criminal or read the licence plate of a car. Since surveillance cameras often employ cheap LR sensors, image upscaling, specifically SR, can enlarge these parts and help the investigation. In remote sensing (e.g., LANDSAT), SR can increase the level of detail of the target and facilitate object detection and classification. Finally, in medical imagining, such as magnetic resonance imaging and computed tomography, SR can be used to enhance the image and thereby support the physicians in their diagnosis.

In some of these applications, like medical and satellite imaging and video applications, it is possible to acquire multiple frames of the same scene, enabling the use of multi-frame SR techniques (Section 3.2.4).

3.1.2 Observation models

Image upscaling is an inverse problem, where the task is to recover an HR image from observed data, i.e., acquired single LR image or multiple LR images. Model-based methods (see Fig. 3.2) assume that an LR image (or images) is obtained through a conventional imaging system, which inevitably causes certain loss of resolution. An HR image is assumed to be ideal undegraded image sampled without frequency aliasing from a continuous scene. This ideal image is only hypothetical, i.e., it does not exist in practice.

In image upscaling from a *single* image, the HR image f and the

observed (measured) LR image g are related through the observation (acquisition) model shown in Fig. 3.3. The HR image f is of size $N = z_1N_1 \times z_2N_2$ and the LR image g is of size $M = N_1 \times N_2$, meaning it is related to the ideal HR image via scaling factors z_1 in horizontal and z_2 in vertical direction, although these are often taken to be the same, $z = z_1 = z_2$. According to the observation model, the scene is first blurred, then downsampled and finally noise is added to create the observed LR image. This is usually described in the matrix form as

$$\mathbf{g} = \mathrm{DB}\mathbf{f} + \mathbf{n}, \tag{3.1}$$

where $\mathbf{g} \in \mathbb{R}^M$ and $\mathbf{f} \in \mathbb{R}^N$ are the vector formulations (assuming raster-scan order) of the LR image g and the HR image f, respectively, $\mathrm{D} \in \mathbb{R}^{M\times N}$ is the downsampling matrix, $\mathrm{B} \in \mathbb{R}^{N\times N}$ is the blur operator and $\mathbf{n} \in \mathbb{R}^M$ is the additive noise. Alternatively, blurring and downsampling can be unified in one linear degradation operation matrix $\mathrm{A} = \mathrm{DB}$, where $\mathrm{A} \in \mathbb{R}^{M\times N}$. Thus the model can be described as

$$\mathbf{g} = \mathrm{A}\mathbf{f} + \mathbf{n}. \tag{3.2}$$

This observation model models the degradations that occur during the process of acquisition of a digital image due to the imperfection of an imaging system. Conventionally, these degradations include blur, noise and aliasing effects, illustrated in Fig. 3.3. Blurring may be caused by an optical system (e.g., out of focus, diffraction limit, atmospheric blur, etc.), relative motion between the imaging system (camera) and the original scene (the so-called motion blur), and the point spread function (PSF) of the LR sensor. The characteristics of the blur are usually assumed to be known.[1] Downsampling is caused by insufficient pixel density and results in aliasing effects. In practice, downsampling is preceded with an anti-aliasing filter in order to avoid aliasing effects. However, a filter that avoids any aliasing and blurring introduces ringing artefacts, thus a compromise must be made between aliasing, ringing and blur artefacts. Therefore, it is often assumed that in a real camera the aliased components are just attenuated to some degree. Finally, noise can appear in the sensor due to analogue circuitry or during transmission [Park 03].

The model described above does not consider compression when storing an image in a lossy compressed format (e.g., JPEG), which can result in quantization and block artefacts. Furthermore, the model could also include other operations, which are built in the camera, such as gamma and colour correction, contrast enhancement, sharpening, demosaicing, etc. However, these are manufacturer-specific and thus not considered in a general image upscaling approach [Luong 09].

At this point, we also introduce the observation model used for image upscaling from multiple LR images, which are sub-pixel shifted and aliased.

[1] Some methods perform blind SR by also identifying blur properties during reconstruction procedure.

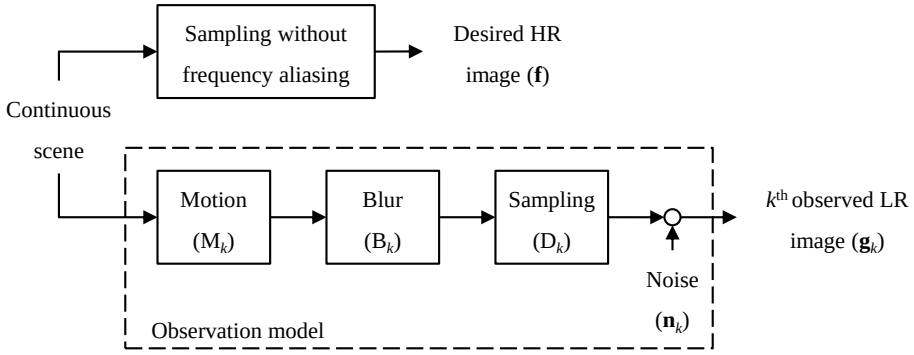


Figure 3.4: Observation model relating HR image with *multiple* observed LR images.

Basically, this model, depicted in Fig. 3.4, relates each LR image $\mathbf{g}_k$, $k = 1, \ldots, N_i$, with the desired, ideal HR image [Park 03]. The difference from the model in Fig. 3.3 is that it assumes motion of the scene, such as global or local translation, rotation, etc. The motion can be described as a warp matrix M_k, leading to the following representation of the model:

$$\mathbf{g}_k = \mathrm{D}_k\mathrm{B}_k\mathrm{M}_k\mathbf{f} + \mathbf{n}_k = \mathrm{W}_k\mathbf{f} + \mathbf{n}_k, \quad k = 1, \ldots, N_i. \tag{3.3}$$

The matrix $\mathrm{W}_k = \mathrm{D}_k\mathrm{B}_k\mathrm{M}_k$ of size $M \times N$ represents the overall degradation matrix and includes motion, blurring and downsampling. Note also that, in general, all degradations can be considered to be different for each LR image, hence the index k for each matrix. This model is used for multi-frame SR approach, which will be described in more detail in Section 3.2.4.

3.1.3 Image upscaling as a regularization problem

Image upscaling, as many other problems in image restoration and computer vision, is ill-posed. Ill-posed problems do not fulfil Hadamard's postulates of well-posed problems, which state that the solution has to exist, be unique and depend continuously on the initial data [Poggio 85]. This is because the number of LR images can be insufficient for the good reconstruction of an HR image, especially in the case of image upscaling from a single image. Furthermore, blur operators can be ill-conditioned, noise is present and amplified by deblurring, etc.

A straightforward way to solve the problem from Eq. (3.2), would be to choose the $\mathbf{f}$ that maximizes the likelihood function $P(\mathbf{g}|\mathbf{f})$:

$$\hat{\mathbf{f}}_{ML} = \arg\max_{\mathbf{f}} P(\mathbf{g}|\mathbf{f}). \tag{3.4}$$

$\hat{\mathbf{f}}_{ML}$ is called the maximum likelihood estimate (MLE). Since noise is typically modelled as zero-mean additive white Gaussian noise (AWGN) with standard

deviation σ_n and $\mathbf{f}$ is assumed to be known [Elad 09], the measurement vector $\mathbf{g}$ is also a Gaussian random vector with the shifted mean, thus the likelihood function becomes

$$P(\mathbf{g}|\mathbf{f}) = \frac{1}{(2\pi)^{M/2}\sigma_n^M} \exp\big(-\frac{1}{2\sigma_n^2}\|\mathbf{g} - \mathrm{A}\mathbf{f}\|_2^2\big), \tag{3.5}$$

and the MLE is

$$\hat{\mathbf{f}}_{ML} = \arg\max_{\mathbf{f}} P(\mathbf{g}|\mathbf{f}) = \arg\min_{\mathbf{f}} \|\mathbf{g} - \mathrm{A}\mathbf{f}\|_2^2, \tag{3.6}$$

where $\|\cdot\|_2$ denotes the L_2-norm. MLE chooses the HR image based only on the measurements, i.e., the observed LR image [Elad 09]. The likelihood is also referred to as the *data fidelity term* since it enforces that the blurred and downsampled version of the HR image resembles as much as possible the input LR image. MLE, however, is severely under-constrained, especially when performing image upscaling from a single image, because the number of constraints induced by the LR image is smaller than the number of unknowns in the HR image. Hence, there are infinitely many solutions to the problem.

To overcome this issue, prior knowledge about the HR image can be used as an additional source of information, leading to the concept of regularization [Jain 89]. Regularization is a way of introducing numerical stability into inverse ill-posed problems by narrowing the solution space to a sub-space where solution is well-defined. The simplest regularization, known as Tikhonov regularization [Tikhonov 77], requires the penalty function to be convex and, therefore, having a unique solution. It does so by adding a smoothness constraint (or regularization term) $\|\mathrm{R}\mathbf{x}\|_2^2$ to the data fidelity term from Eq. (3.6), leading to the regularized solution

$$\hat{\mathbf{f}}_R = \arg\min_{\mathbf{f}} \big(\|\mathbf{g} - \mathrm{A}\mathbf{f}\|_2^2 + \lambda\|\mathrm{R}\mathbf{f}\|_2^2\big), \tag{3.7}$$

where R is generally a high-pass filter and λ is the Lagrange multiplier, referred to as the regularization parameter. This type of regularization enforces spatial smoothness uniformly over the HR image by penalizing the amount of high frequencies in the HR image, under the assumption that images are naturally smooth with limited high-frequency activity. However, this smoothness prior typically yields over-smoothed results. λ can be used to control the influence of data fidelity and smoothness (regularization) term. Larger values of λ cause smoother solution, which is useful when the number of observed LR images is small and/or the amount of noise in the LR image is relatively high. On the other hand, in the case of the multi-frame approach (Section 3.2.4), when large number of LR images is available and the amount of noise is small, smaller values of λ yield better results.

One can gain much more from regularization than just numerical stability by looking at regularization from a Bayesian point of view. The Bayesian approach provides robustness and flexibility in modelling prior knowledge about

the image, which has led to the evolution of image priors over the years in different restoration problems and improved the quality of the result. Excellent review of image priors can be found in [Elad 09], and we will review some of the priors used for image upscaling in Sections 3.2.3, 3.2.4 and 3.2.5.

The Bayesian approach aims at estimating the HR image based on the posterior probability $P(\mathbf{f}|\mathbf{g})$ of the HR image given the observed LR image, thus $\mathbf{f}$ is now assumed to be random as well. This posterior probability can be expressed via the Bayes rule as $P(\mathbf{f}|\mathbf{g}) \propto P(\mathbf{g}|\mathbf{f})P(\mathbf{f})$ (see also Section 2.2.4). Typically, the HR image $\mathbf{f}$ is estimated by maximizing a posteriori (MAP) probability, known as the MAP approach:

$$\hat{\mathbf{f}}_{MAP} = \arg\max_{\mathbf{f}} P(\mathbf{f}|\mathbf{g}) = \arg\max_{\mathbf{f}} P(\mathbf{g}|\mathbf{f})P(\mathbf{f}). \tag{3.8}$$

The prior term $P(\mathbf{f})$ describes the probability density function (PDF) of the HR image and can be represented in the exponential form via Gibbs distribution as

$$P(\mathbf{f}) = \alpha \exp\big(-\beta E(\mathbf{f})\big), \tag{3.9}$$

where the constant α is the normalization factor and $E(\mathbf{f})$ is a non-negative energy function (see also Section 2.2.2). Then, in the case of AWGN, the likelihood term $P(\mathbf{g}|\mathbf{f})$ is given by the Eq. (3.5) and the MAP estimate of the HR image becomes:

$$\hat{\mathbf{f}}_{MAP} = \arg\max_{\mathbf{f}} P(\mathbf{g}|\mathbf{f})P(\mathbf{f}) = \arg\min_{\mathbf{f}} \big(\|\mathbf{g} - \mathbf{A}\mathbf{f}\|_2^2 + \beta E(\mathbf{f})\big). \tag{3.10}$$

We can see that the MAP approach introduces regularization to the solution. The difference from Eq. (3.7) is that the additional constraint now has a probabilistic meaning, thus Eq. (3.7) is a special case of Eq. (3.10). In order to use the Bayesian approach, one needs to specify the energy function $E(\mathbf{f})$, which should describe the image behaviour.

3.2 Image upscaling: an overview

As depicted in Fig. 3.2, we classify the methods into two main groups: interpolation and model-based methods. Interpolation methods treat image upscaling as an interpolation problem, assuming that the continuous scene is sampled with Dirac pulses. They can be further divided into two categories: linear and adaptive. Linear methods (Section 3.2.1) estimate a missing HR pixel as a linear combination of LR pixels, while adaptive methods (Section 3.2.2) adapt interpolation coefficients to image content (e.g., edges vs. smooth areas). Model-based methods treat image upscaling as an inverse problem (Section 3.1.3). We divide this group of methods into reconstruction-based (Section 3.2.3), multi-frame (Section 3.2.4) and example-based (Section 3.2.5) methods. Reconstruction-based methods enforce a reconstruction constraint (or

data fidelity) on the HR image together with some prior knowledge about the image in order to minimize the artefacts introduced by linear interpolation. Multi-frame SR reconstructs the HR image from multiple LR images of the same scene, which are shifted by sub-pixel shifts and contain frequency aliasing. Example-based SR methods attempt to fill in the missing high frequencies by searching for highly similar examples (usually small patches of pixel values) in the external database that also contains HR information, or by exploring the self-similarity within and across scales of the input LR image.

3.2.1 Linear interpolation

Linear interpolation methods aim at approximating the underlying continuous function from which the pixels of the LR image have been sampled, i.e., to perform discrete-to-analogue conversion based on the discrete samples. This process can formally be regarded as a convolution of the (discrete) image with the continuous interpolation kernel h_{2D}:

$$q(u,v) = \sum_a \sum_b q(a,b) h_{2D}(u-a, v-b), \tag{3.11}$$

where $u, v \in \mathbb{R}$ and $a, b \in \mathbb{N}$. Since convolution is a linear operation, this class of methods is called linear (or non-adaptive) interpolation. Usually, symmetrical and separable kernels are employed, $h_{2D}(u,v) = h(u)h(v)$, to decrease the computational complexity and make the implementation in multiple dimensions straightforward. Practically, this means that interpolation of two-dimensional images is split into horizontal and vertical interpolation, which are performed consecutively.[2] Examples of typical interpolation kernels are shown in Fig. 3.5.

The result of image interpolation depends greatly on the choice of the interpolation kernel, which is usually a trade-off between complexity and image quality. The optimal interpolation kernel is a sinc function, but because of its infinite support and slow decay, it is rarely used in practice [Unser 00]. There is a vast number of sinc-approximating kernels, i.e., kernels trying to resemble the sinc function, that have been used in image processing over the years. The simplest method is *nearest-neighbour* interpolation, which assigns the interpolated pixel the value of the pixel that is closest to it. A slightly better approach that produces smoother results is *bilinear* interpolation, a 2D extension of linear interpolation. This method determines the interpolated value as a weighted average of known pixel values in a 2×2 neighbourhood surrounding the location of the unknown pixel. The weights are computed based on the distance of the unknown pixel to the known ones. Even better results can be obtained with *bicubic* interpolation, which makes use of a 4×4 neighbourhood and produces sharper results with less interpolation artefacts (see Fig. 3.6). Next to these common methods, higher-order interpolation methods exist. They take more neighbouring pixels into consideration and thus

[2] This is valid for images acquired on Cartesian grids, which is often the case for images and videos. Other types of sampling grids, e.g., hexagonal lattices, also exist.

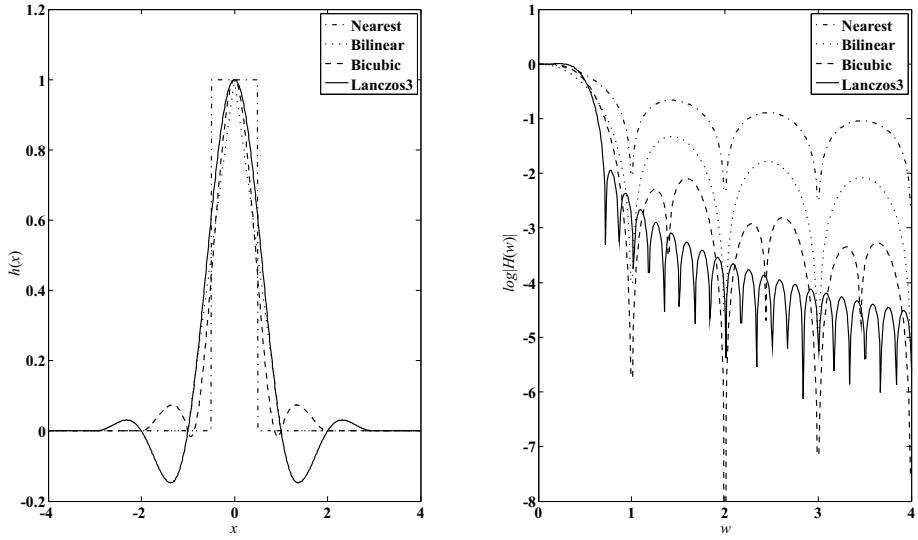


Figure 3.5: Typical interpolation kernels in the spatial (left) and the frequency spectrum domain (right).

are computationally more intensive, but often without producing substantially higher quality. Furthermore, a common choice of the interpolation kernel is the windowed sinc interpolation kernel, which is a sinc function multiplied by a windowing function with a limited spatial support. Many possible windowing functions exist, e.g., Lanczos, Blackman-Harris, Hamming, etc. For a detailed overview of windowing functions see, e.g., [Meijering 01]. Finally, there is also a group of generalized convolution kernels, among which the most popular member is the B-spline family [Unser 99]. For a detailed survey of linear interpolation methods see [Lehmann 99, Meijering 01].

Despite the variety of linear interpolation methods, they still have several disadvantages, such as not restoring missing high frequencies, not dealing with noise and quantization and finally introducing a number of artefacts. The most common artefacts are *staircase* (or *jaggy*) artefacts, *blur* and *ringing* artefacts, illustrated in Fig. 3.6. Jaggy artefacts are caused by non-ideal nature of the interpolation kernel, which has sidelobes and ripples present in the stop-band (see Fig. 3.5). Hence, frequency components of the replicated spectra are badly suppressed. They are most prominent in Fig. 3.6(b), i.e., in the result of nearest neighbour interpolation, but they are also present in bilinear and bicubic interpolation results (Figs. 3.6(c) and (d), respectively). Blur artefacts are demonstrated as blurring the sharp edges, which is also caused by the non-ideal interpolation kernel that attenuates (high) frequencies in the pass-band (see Figs. 3.6(c), (d) and (e)). Finally, ringing artefacts are represented as ghost repetitions or *halos* in the uniform areas near edges (Fig. 3.6(e)). This is due to the band-limited interpolation kernel that stops the high-frequency

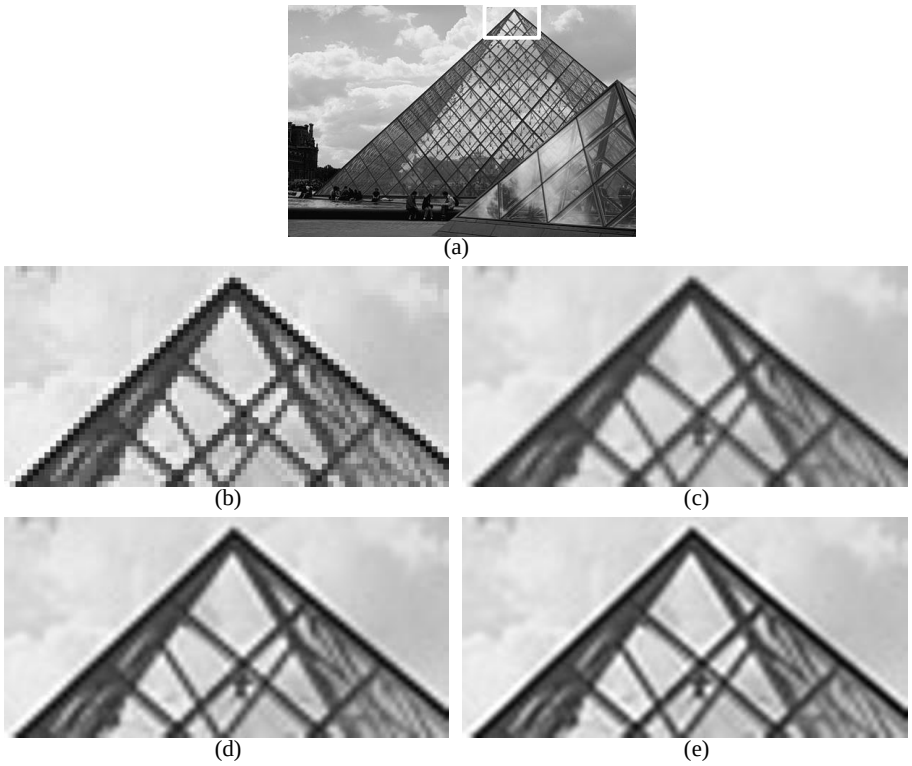


Figure 3.6: Results of linear interpolation methods from Fig. 3.5: (a) full original image with the marked cropped part, (b) result of nearest neighbour interpolation, (c) result of bilinear interpolation, (d) result of bicubic interpolation, and (e) result of Lanczos3 interpolation.

components abruptly.

3.2.2 Adaptive interpolation

The artefacts around edges induced by linear interpolation methods have prompted a development of image upscaling methods that aim at adapting interpolation by imposing more accurate models and by incorporating prior knowledge. Hence, they are referred to as *adaptive* interpolation methods. These methods are also *non-linear*, since they no longer perform a simple convolution with an interpolation kernel. They are able to produce visually more pleasing results with sharper edges and less artefacts (previously mentioned in Section 3.2.1). Adaptive methods can be broadly divided into two groups: *edge-directed* and *restoration-based* interpolation methods.

The main idea of **edge-directed interpolation** is to exploit the property of an image called *geometric regularity* [Mallat 98], which means that

image intensity field varies little along the edge orientation, while across edge orientation significant changes can occur. Since edges are important features in natural images, exploiting geometric regularity is a valid approach for maintaining edges' sharpness and eliminating artefacts. Therefore, the main idea is to make sure that the image is not smooth perpendicular to edges, but smooth parallel to edges.

Different methods exploit geometric regularity in different manner. One group of methods [Jensen 95, Allebach 96, Yu 01, Su 04, Battiato 02, Muresan 05, Wang 07] attempts to detect the location or orientation of the edges explicitly and adjust the interpolation coefficients accordingly, i.e., they modify linear interpolation to prevent interpolation across edges. The problem with these methods is that the explicit detection of natural edges is difficult because they can appear as blurred and/or noisy. This implies that the characteristics of edges, such as width, location and direction, are also difficult to estimate. Furthermore, in some of the methods, e.g., [Jensen 95, Battiato 02, Muresan 05], edge orientation is quantized into a finite number of choices (e.g., horizontal, vertical, diagonal), which leads to a less accurate edge model, but improves the computational efficiency.

More accurate results can be obtained with implicit edge-directed interpolation methods [Li 01, Muresan 01, Muresan 04, Li 08], where edge information is not estimated explicitly, but rather the principle of geometric regularity is built into the algorithm itself. These methods yield visually significantly better interpolation results than linear interpolation methods, especially when considering jaggy artefacts, because they are able to tune interpolation coefficients to match an arbitrary edge orientation. However, the sharpness of edges could still be improved and it makes textured areas look unnatural, although improvements have been made in this respect in [Muresan 04, Li 08]. The most popular method from this group is the new edge-directed interpolation (NEDI) [Li 01], which uses the duality between LR and HR covariance to estimate the HR image. In other words, the covariance between neighbouring pixels in a local window in the LR image is used to estimate the covariance between neighbouring pixels in the HR image.

Among edge-directed interpolation methods, we ought to mention the methods based on multi-resolution analysis, mainly in the wavelet domain [Chang 95, Carey 99, Kinebuchi 01]. These methods estimate the wavelet coefficients of the fine scale based on the the wavelet coefficients of the coarser scale and some assumed model between them. For example, methods in [Chang 95, Carey 99] model the relationship between exponential decay of coarse and fine wavelet scales [Mallat 92], while the method in [Kinebuchi 01] uses Gaussian mixture models and hidden Markov trees [Romberg 01].

The goal of the **restoration-based interpolation** methods is to minimize the artefacts in the roughly initialized HR image (e.g., obtained by linear interpolation) through an iterative optimization process, by putting constraints on the HR image. The most popular approach is based on partial differential equations (PDEs) [Morse 01, Jiang 02, Luong 05]. PDEs describe

the evolution of curves, surfaces or vector fields, and are often used in other image processing fields (see, e.g., Section 4.2). In [Morse 01,Luong 05], smoothness of level curves or isophotes is enforced by minimizing their curvature, while in [Luong 05], constrained adaptive contrast enhancement is additionally used to sharpen the edges. Typically, PDE-based methods are iterative, but by using some approximations PDE can be solved in a one-pass discrete form to improve the efficiency, as suggested in [Jiang 02, Luong 05]. In general, these methods yield promising results since they are able to eliminate jaggy artefacts and produce sharp edges.

Another group of restoration-based interpolation methods use a *projection-onto-convex-sets* (POCS) scheme to impose constraints on the HR image [Gerchberg 74, Papoulis 75, Ferreira 94, Ratakonda 98]. The solution, i.e., the HR image, is restricted to belong to a convex set, which is defined as a set of vectors satisfying certain properties, such as fidelity to the observed data, positivity, smoothness, etc. Multiple constraints result in having multiple convex sets. The solution lies in the intersection of these sets and can be found in an iterative procedure by projecting it alternatively onto these sets.

3.2.3 Reconstruction-based methods

Reconstruction-based methods treat image upscaling as an inverse problem of the degradation process. They assume that LR images are obtained by smoothing and downsampling HR scenes with low-quality image sensors, as explained earlier in Sections 3.1.2 and 3.1.3. The methods that we review in this subsection are related only to image upscaling from a single input LR image.

Similarly to restoration-based interpolation methods, reconstruction-based methods aim at minimizing the artefacts in the roughly initialized HR image through an iterative optimization process. However, unlike interpolation methods, they assume that LR images are corrupted by noise, blur, etc. Their main requirement is that the data fidelity term (here referred to as the *reconstruction constraint*), introduced in Eq. (3.5), is satisfied. Iterative back-projection (IBP) [Irani 91] was proposed to minimize reconstruction error efficiently through an iterative process for multi-frame SR, thus it will be explained in more detail in Section 3.2.4. It can also be used for image upscaling from a single image, but results are insufficiently good because of the ill-posed nature of the problem.

As explained in Section 3.1.3, image priors are employed to regularize the inverse problem by introducing an additional constraint on the HR image. Two widely used image priors are image smoothness and edge smoothness prior. Image smoothness prior enforces smoothness across the whole image, thus the resulting HR image contains blurred edges and textures. Edge smoothness prior, on the other hand, has edge preserving property because it simultaneously enforces the smoothness of edges along edge direction and prevents smoothing orthogonally to edge direction, which is consistent with human perception [Dai 09]. Similar strategy was employed for adapting inter-

polation weights in edge-directed interpolation and imposing prior knowledge in restoration-based interpolation (Section 3.2.2).

Edge smoothness prior was used in many image upscaling techniques [Rabaud 05, Aly 05, Tai 06, Dai 09, Luong 07, Fattal 07, Sun 08]. How exactly the edge smoothness constraint is imposed, and based on which features, differs greatly between methods. For example, in [Aly 03, Aly 05], total variation (TV) is proposed for regularization and PDEs are employed to obtain the solution, while in [Tai 06], HR curves are inferred by multi-scale tensor voting [Medioni 00]. An interesting way to impose edge-preserving constraints is to exploit the statistical dependency between edge features at different resolutions [Fattal 07,Sun 08]. The two methods differ in the proposed edge statistics and in the edge features from which they draw the statistics.

Other than image smoothness and edge smoothness prior, also some other priors have been used for image upscaling, like sparse derivative prior [Tappen 03] and colour prior [Luong 07]. A very promising approach is also to define the prior based on examples, rather than guessing a mathematical expression. We will review this approach in more detail in Section 3.2.5. Regardless of which prior is used, reconstruction-based methods also have the requirement to satisfy the reconstruction constraint, resulting in an optimization problem from Eq. (3.10).

3.2.4 The multi-frame approach

3.2.4.1 Main concepts

Unlike the previously reviewed techniques, methods using the multi-frame approach are able to recover the missing or degraded high-frequency components in the HR image and, therefore, are regarded as SR methods. This characteristic relies on the availability of multiple LR images of the same scene, which, in general, must be sub-pixel shifted and contain frequency aliasing. If there is only an integer shift, then each LR image contains the same information, thus no new information is available for the HR image reconstruction. On the other hand, if there is no aliasing, the observed data contains only exact band-limited information, which makes the recovery of missing high frequencies impossible. Applications where the acquisition of such multiple LR images is possible include medical imaging, satellite imaging and video applications.

The multi-frame approach assumes the observation model described in Section 3.1.2 and depicted in Fig. 3.4. In order to recover the HR image, most of multi-frame methods consist of three stages [Park 03]: registration, interpolation and restoration (Fig. 3.7). Registration involves estimation of motion, i.e., shifts with sub-pixel accuracy, of each LR image relative to one reference LR image. Accurate sub-pixel motion estimation is very important for the success of these algorithms, so attempts have been made to use accurate registration based on robust motion models [Borman 99] or to include the registration error in the subsequent reconstruction procedure [Ng 02, Lee 03]. Registration is followed by interpolation, also called *data fusion*, which interpo-

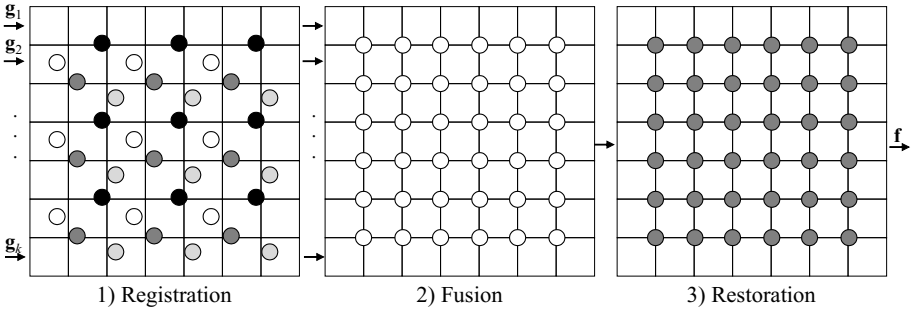


Figure 3.7: Illustration of the three stages of multi-frame SR methods.

lates the non-uniformly spaced composite of LR images to uniformly spaced HR grid. For this purpose, a simple method such as bilinear interpolation could be used, although vast number of methods have been proposed in literature (see [Luong 09] for a detailed overview). Finally, image restoration is applied to remove blur and noise.

3.2.4.2 An overview of algorithms

Literature contains many multi-frame SR methods, which differ in the domain in which they are operating (spatial or frequency), the observation model assumed, the type of the reconstruction method, etc. Excellent reviews of different approaches can be found in [Park 03, Borman 04]. According to [Park 03], multi-frame SR methods can be divided into five big groups: non-uniform interpolation methods, frequency-domain methods, regularized approaches, projection onto convex sets (POCS) and other SR reconstruction approaches.

The **non-uniform interpolation approach** [Ur 82, Shah 99, Alam 00, Nguyen 00, Farsiu 04, Luong 09] is the most intuitive method for SR reconstruction, because it directly follows the three stages from Fig. 3.7. The advantage of this approach is that it is relatively computationally simple, making it a candidate for real-time applications. On the other hand, the degradation model is limited to having the same noise and blur characteristics for all input LR images. Furthermore, the whole method is not optimal because the restoration step ignores errors that may occur during interpolation [Park 03].

The **frequency-domain approach** [Tsai 84, Kim 90, Kim 93, Rhee 99] makes explicit use of the aliasing present in the LR images to reconstruct the HR image. As formulated in [Tsai 84], this is achieved through the shifting property of the Fourier transform, the aliasing relationship between the samples of the continuous Fourier transform of the original HR image and discrete Fourier transform of the observed LR image, and the assumption that the HR image is band-limited. These basic principles make frequency-domain approach theoretically simple and also convenient for parallel implementation, but on the other hand, the observation model considers only global translation and linear space-invariant blur. Furthermore, prior knowledge cannot be added

as regularization since it is difficult to formulate it in the frequency domain.

The **regularized approach** uses the regularization framework described in Section 3.1.3, either in a deterministic way by introducing numerical stability in the HR solution [Hong 97, Hardie 98, Bose 01] or via the Bayesian approach [Tom 95, Schultz 96, Hardie 97, Zibetti 07]. It is based on the same concepts like reconstruction-based approach from Section 3.2.3, but the inverse problem is more constrained since multiple images are available and, therefore, mostly a generic smoothness prior is used. The advantage of these methods is the flexibility in modelling noise characteristics and prior knowledge about the HR image, but also the possibility of simultaneous motion and HR image estimation.

The **POCS approach** [Stark 89,Patti 97,Eren 97,Patti 01], also used for single-image upscaling as mentioned already in Section 3.2.2, is the alternative approach for including prior knowledge into the image upscaling process. In the multi-frame case, the method estimates the registration parameters and then performs interpolation and restoration simultaneously. The advantage of this approach is its conceptual simplicity, powerful spatial domain observation model and convenient inclusion of prior information. However, the method does not provide a unique solution, in addition to having a slow convergence and a high computational cost. In order to overcome the problem of non-unique solution, a hybrid method was introduced [Elad 97], which combines the MLE or MAP approach with the POCS constraints.

Other multi-frame methods include iterative back-projection (IBP) [Irani 91], adaptive filtering [Elad 99b, Elad 99a] and the methods based on the motionless approach [Elad 97, Rajan 01b, Rajan 01a, Rajan 02, Joshi 02]. The motionless approach does not require existence of motion between LR images, but rather uses other cues to reconstruct the HR image, such as differently blurred LR images [Elad 97, Rajan 01b, Rajan 02], photometric cues [Rajan 01a] and zoom [Joshi 02]. The IBP method is frequently used for comparison and employed in reconstruction-based and example-based approaches to impose that the reconstructed HR image resembles the input LR image. Therefore, we explain it here in more detail. IBP updates the current estimate of the HR image by back-projecting the residual error (difference) between simulated LR images, i.e., LR images obtained by applying the overall degradation matrix W_k from Eq. (3.3) on the current HR estimate, and observed LR images:

$$\hat{\mathbf{f}}^{(t+1)} = \hat{\mathbf{f}}^{(t)} + \sum_{k=1}^{N_i} \mathrm{W}_k^{BP}(\mathbf{g}_k - \mathrm{W}_k\hat{\mathbf{f}}^{(t)}). \tag{3.12}$$

Note that if $N_i = 1$, IBP can be applied as an image upscaling algorithm from a single input image. There are a few versions of IBP algorithm [Peleg 87, Irani 91, Zomet 01, Irani 93], of which the one from [Irani 91] is the most often used. It calculates the error as the sum of squared differences between simulated and observed LR images and updates the HR estimate via the back-projection kernel, i.e., W_k^{BP} ensures that the HR pixel is updated based on all contributing LR pixels according to the PSF of the observation model.

Although this algorithm is simple, it suffers from a series of problems, such as non-uniqueness of the solution, difficulty in choosing the back-projection kernel, dependence on the initial guess, etc.

3.2.5 The example-based approach

3.2.5.1 Main concepts

The example-based approach [Freeman 00], just like the multi-frame approach, is an SR method because it aims at recovering missing HR details (high frequencies) that are not present in the LR image. However, unlike the multi-frame approach, which assumes that this information is available across multiple LR images in an aliased form, example-based methods typically try to retrieve these missing high frequencies from a training database based on the similarity between the input LR image and training examples. This database contains pairs of HR/LR examples and is used to learn the correspondences between HR and LR images, during the so-called *learning phase*. The learning phase also reduces the number of examples to some manageable number, thus simplifying the following *reconstruction* phase, where learned correspondences are applied to the new input LR image in order to obtain its HR version.

The example-based approach has become increasingly popular over the last decade for its ability to overcome the limitations of the multi-frame approach. These limitations are the following: 1) the multi-frame approach requires a sequence of degraded LR images in order to reconstruct an HR image, and 2) it is limited in practice to the magnification factor smaller than two [Baker 02, Lin 04]. Example-based SR is able to reconstruct an HR image from a *single* input LR image with much higher magnification factor (in preliminary work of Lin et al. [Lin 08], the limit for the magnification factor for learning-based approaches is estimated at 10, although it highly depends on the database). Furthermore, the quality of the result is very encouraging, both for the reconstruction of edges and textured areas.

However, still some problems remain. First, example-based methods involve storing and searching large databases, especially when applied to natural images [Freeman 00, Freeman 02, Wang 05, Xiong 09], although some attempts have been made to constrain the search to a single input texture [Tai 10] or texturally relevant segments in the database [Sun 10, HaCohen 10]. Searching the database can be avoided by using it only to learn the interpolation functions [Tappen 03, Tappen 04, Kim 08], but still this external database is necessary. Additionally, it is not guaranteed that the database contains the true HR details, which may cause the so-called "hallucination" effect resulting in added artefacts. Furthermore, this database needs to be large enough to provide good results, which makes learning or searching computationally more demanding. These problems recently prompted the development of *single-image example-based* methods [Ebrahimi 07, Suetake 08, Glasner 09, Luong 10, Yang 10b, Freedman 11], which explore image self-similarity within and across scales of the input LR image and thus do not require an external database.

Figure 3.8: Patch pairs in external database of [Freeman 00] (figure taken from the same paper): LR patches (top) and HR patches (bottom).

Let us first consider how to represent examples, although this is still an opened research question [Elad 09]. Examples are often represented as small image patches of raw pixel values (e.g., in [Datsenko 07, Yang 08, Elad 09]), where sometimes the mean value is subtracted to make the patch representation more general (i.e., to increase the number of possible matches). Another option is to use patches of extracted features, such as high frequencies [Freeman 00, Freeman 02], derivatives [Chang 04, Tai 10], principal component analysis (PCA) coefficients [Wang 05], etc. In [Xiong 09], it was shown that feature enhancement of LR images, as a combination of interpolation with pre-filtering and non-blind sparse prior deblurring, can improve matching, and, therefore, SR results. Since example representation is usually based on patches, example-based approaches are often referred to as *patch-based*.

The external database is usually formed by collecting HR/LR patch pairs from training HR images and their degraded LR versions, respectively. For example, in the seminal work of Freeman et al. [Freeman 00], HR patches are extracted from the difference images between the original HR images and their degraded versions, obtained by blurring and downsampling HR images and interpolating these back to the original sampling resolution. Therefore, the difference patches represent the high-frequency detail removed by the degradation process. LR patches are taken from the degraded HR images, whose low frequencies are additionally removed to create a band-pass "image". Finally, both difference and band-pass images are contrast normalized by the local contrast of the input band-passed image (see Fig. 3.8). Some methods use pixel-based features as examples instead of patches. For example, in [Baker 02], the database is formed as a collection of pixel-based sets of features extracted from derivative pyramids of training images. On the other hand, as mentioned earlier, the input image itself can be used as a source of examples based on the observation that small image patches tend to recur many times inside a natural image, both within the same scale and across different image scales. This image property is called self-similarity and it was statistically analysed in [Glasner 09], while in [Zontak 11] a parametric quantification of this property was derived. Redundancy of patches within the same scale was previously successfully explored for texture synthesis [Efros 99], image inpainting (e.g., [Criminisi 04]) and image denoising via non-local means (NL-means) [Buades 05]. Redundancy across scales, on the other hand, provides HR/LR example pairs, thus enabling example-based SR from the input LR image itself, without using external training database. Such an "internal" database has a limited size, but its content is more relevant to the target HR image, thus the influence of the

"hallucination" effect is decreased.

Elad and Datsenko [Elad 09] reviewed example-based methods from a regularization point of view (see Section 3.1.3), in the sense that examples can help in defining the image prior, rather than arbitrarily or intuitively choosing a PDF. They divided the methods for inverse problems in general (not only SR) based on how to use examples: directly in the reconstruction result, plugged into regularization expression or by training regularization parameters. We make a different classification by addressing learning and reconstruction phase of each group, and ultimately, we divide the methods into five groups: methods using examples directly, methods using examples to build a regularization expression, methods using examples to learn interpolation functions, methods using sparse representations and single-image example-based methods.

3.2.5.2 An overview of algorithms

Methods using examples directly [Freeman 00, Freeman 02, Sun 03, Chang 04, Wang 05, HaCohen 10] search the database for the nearest neighbour (or neighbours)[3] of each LR patch of the input image during the learning phase. The nearest neighbour search is part of the online reconstruction, thus should be performed efficiently. For speed-up, PCA representation of image patches was proposed in [Freeman 00, Wang 05], as well as fast approximate search methods, such as tree-based, approximate nearest neighbour search [Nene 97, Freeman 02], and locality sensitive hashing [Gionis 99, Wang 05]. Each found nearest neighbour (i.e., LR patch) has a corresponding HR patch in the database. These HR patches may be regarded as samples from the posterior $P(\mathbf{f}|\mathbf{g})$ (see Eq. (3.8)). The goal of the reconstruction phase is then to combine the HR patches in order to form the HR image. For this step, different solutions are proposed. In [Freeman 00, Sun 03, Wang 05], an MRF model is defined over the HR image to impose the global agreement of HR patches and then inference is performed to choose one HR patch per location (for the details of MRF modelling see Chapter 2).[4] Additionally, the method in [Wang 05] considers the problem of blind SR, when the PSF parameter of the imaging system is unknown. A simpler reconstruction can be achieved with a one-pass algorithm proposed in [Freeman 02], which selects an HR patch based on its LR patch and neighbouring HR patches that are already selected. The method from [Chang 04] uses local linear embedding [Roweis 00] from manifold learning by computing the reconstruction weights that minimize the reconstruction error of representing input LR patch with its found neighbours. Those weights are then used to linearly combine corresponding HR patches of the neighbours from the database. More recently, HaCohen et al. [HaCohen 10] proposed to improve the patch-based model for better treatment of textures by segment-

[3] The nearest neighbours are the most similar patches of the query patch, because patches can be regarded as points in a multi-dimensional space, thus patch matching can be transformed to a nearest-neighbour search.

[4] The method from [Freeman 00] will be explained in more detail in Section 3.4.2, since our proposed method is based on it.

ing the input image and matching each segment with user-assistance with one texture from the example texture database, which is then searched for the best-matching patches.

Methods using examples to build a regularization expression [Baker 02, Pickup 03, Datsenko 07, Elad 09, Tai 10, Sun 10] have the same learning phase as the previous group of methods, i.e., for each LR patch or pixel, nearest neighbours are found in the database. However, in the reconstruction phase, instead of using these found examples directly, as such, they build a regularization expression, which forces the proximity between them and the corresponding features of the HR image. In this way, examples serve as an additional constraint, a sort of image prior in a broad sense, that together with a reconstruction constraint, which ensures proximity to the measurements (observed LR image), forms the final optimization problem. This approach is then global, since the regularization expression is defined over the whole image, but the learning phase is still local, because the examples are found patch-per-patch, which enables parallelization and simplification of the algorithms [Elad 09]. Recent trends in this approach involve adding one more constraint to impose edge smoothness, as in [Tai 10, Sun 10]. In this way, the edges of the HR image are forced to be sharp, with minimal ringing or jaggy artefacts, since the quality of edges may be compromised by using the database. Additionally, Sun et al. [Sun 10] proposed the context-constrained approach for the learning phase, in the sense that the patch search is constrained to the texturally similar *segments* from the database, similarly to [HaCohen 10]. They show that such an approach leads to better SR result in textured regions. On the other hand, in [Tai 10], a database consists of only one user-supplied input texture, making the approach highly dependent on its choice.

Methods using examples to learn interpolation functions [Tappen 03, Tappen 04, Kim 08] have a different learning phase during which correspondences are learned between HR/LR patch pairs from the training database. For example, in [Tappen 03, Tappen 04], a small number of linear regression functions are learned with the procedure similar to expectation-maximization, while in [Kim 08], kernel ridge regression is used to find a mapping between HR and LR patches. Once these functions are learned, they are applied to each input LR patch to obtain a set of candidate HR patches for reconstruction. To obtain the final HR image, similarly to [Freeman 00], loopy belief propagation (LBP) [Yedidia 01b] (Section 2.3.5) is used for inference on the MRF model to choose one HR patch per location. These methods do not require searching the huge amount of data in the database during online reconstruction, thus having the potential of being less computationally intensive. However, learning regression functions can be quite computationally demanding [Kim 08].

Many recent methods use **sparse representations** of dictionary elements to reconstruct the HR image [Yang 08, Yang 10b, Wang 10, Zeyde 10]. The main idea is to represent the input LR patch as a sparse combination of elements of some LR dictionary, and then directly use obtained sparse coeffi-

cients to recover the corresponding HR patches from some HR dictionary. The LR dictionary and the HR dictionary have to be coupled so that the LR patch and the HR patch have the same sparse representation. Dictionaries are generated in the learning phase, either by randomly sampling raw patches from the training HR/LR image pairs [Yang 08, Wang 10], or by learning the compact dictionary pairs from training examples [Yang 10b, Zeyde 10]. With this latter approach, the speed of the algorithm can be significantly improved, especially in [Zeyde 10], where K-SVD [Aharon 06] is used for dictionary training, even on the input LR image itself, eliminating the need for the training set.

The last group of methods in our classification are **single-image example-based methods** [Ebrahimi 07, Suetake 08, Glasner 09, Luong 10, Yang 10a, Freedman 11]. The learning phase of these methods typically consists of searching for the nearest neighbours of the input LR patch in the lower scales of the input LR image, and extracting their HR counterparts from the higher scale (e.g., input LR image itself) as candidate HR patches. The method in [Freedman 11] shows that this search can be done in extremely localized regions, where a small scaling factor is achieved by applying specially developed filter banks. Such narrowed search leads to a very efficient method with good results. Other approaches mainly differ in the reconstruction phase. For example, Ebrahimi and Vrscay [Ebrahimi 07] use the NL-means framework and compute the HR image as a weighted average of the found examples. Suetake et al. [Suetake 08] compute an example codebook and use it to estimate the missing high-frequency band in the framework similar to [Freeman 02]. Glasner et al. [Glasner 09] exploit patch redundancy both within and across image scales in order to enable a unified approach, which combines the classical multi-frame SR and example-based SR, while Yang et al. [Yang 10a] use sparse representation via a dictionary, which is trained by enforcing group sparsity constraints.[5] We have also contributed to the development of a single-image example-based SR algorithm that, in addition to these non-local similarities within and across scales, uses kernel regression and sparsity constraints to perform HR image reconstruction [Luong 10].

3.3 Notations and definitions for patch-based models

In this short section, we will introduce the most important notations and definitions related to patch-based methods, which we will use throughout the rest of this thesis.

Let I denote a set of pixel positions in some image g of size $N_1 \times N_2$. Pixel positions are represented by a single index p assuming raster-scan order. If a is a horizontal and b a vertical coordinate, then $p = N_1 b + a$. We define a square mask Ψ as a set of positions p centred at the origin $p = 0$.

[5] The method from [Glasner 09] will be explained in more detail in Section 3.4.1, since our proposed method is based on it.

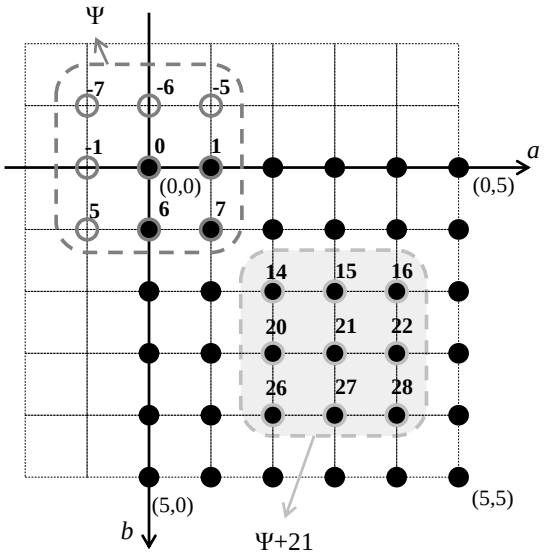


Figure 3.9: Illustration of an $N_1 \times N_2 = 6 \times 6$ image and a 3×3 mask Ψ. Raster coordinates are computed in this case as: $p = N_1 b + a = 6b + a$.

Let $\Psi + p$ denote a translated mask to position p (see Fig. 3.9 for graphical representation). We define a *patch* centred at p' as a set of pixel values $g(p)$, with $p \in \Psi + p'$. A patch is thus specified by a function g, a spatial position p', and a mask shape Ψ.

An important operation in patch-based methods is to compare patches from one or more images by calculating some distance measure between their corresponding pixel values. A common choice is the sum of squared differences (SSD). In order to express this distance, let us first define the translation operation $\mathcal{T}_{p'}$ operating on a function as:

$$\mathcal{T}_{p'}\{g\}(p) \triangleq g(p + p'). \tag{3.13}$$

The braces indicate that the translation operator operates on the function g and not on the function value $g(p)$. We can leave out the braces and write $\mathcal{T}_{p'} g$ when it is clear that the operator operates on a function. We define a norm computed on Ψ of some function h as:

$$\|h\|_\Psi \triangleq \sqrt{\sum_{p \in \Psi} h^2(p)}. \tag{3.14}$$

Now the SSD between two patches in images g' and g'' centred at positions p' and p'', respectively, evaluated on some common shape Ψ can be expressed as $\|\mathcal{T}_{p'}\{g'\} - \mathcal{T}_{p''}\{g''\}\|_\Psi^2$ or $\|\mathcal{T}_{p'} g' - \mathcal{T}_{p''} g''\|_\Psi^2$. This notation means: move the first image g' so that the old position p' is now at the origin. Move the second

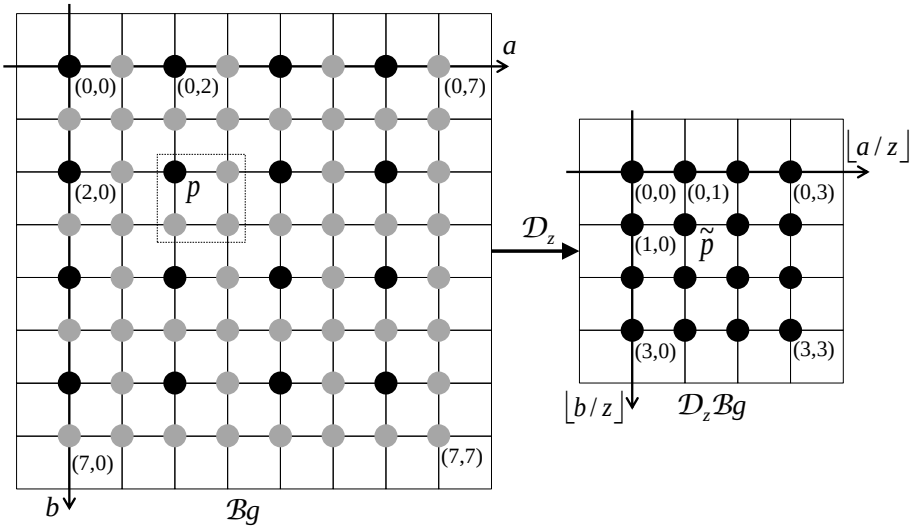


Figure 3.10: Downsampling procedure for $z = 2$. Black circles in the image at higher scale (in the top-left corner of each 2×2 block of pixels in the left image) are kept after downsampling. $p = N_1 b + a = 18$ and $\tilde{p} = \lfloor N_1/z \rfloor \lfloor b/z \rfloor + \lfloor a/z \rfloor = 5$ are corresponding pixel positions in the images at higher and lower scale, respectively.

image g'' so that the old position p'' is now at the origin. Subtract the two images and compute the SSD over a region Ψ centred at the origin.

In the remaining of this section, we also introduce some notations specific to patch-based SR. In SR, let g denote the input LR image and f the ideal (hypothetical) HR image (see Section 3.1.2). We assume that g is related to f by an observation model from Eq. (3.1), but without noise. We represent here the blur matrix B and the downsampling matrix D as the blur operator $\mathcal{B}$ and the downsampling operator $\mathcal{D}_z$, respectively, where z is the integer scaling factor. Hence, we can write $g = \mathcal{D}_z\mathcal{B}f$. To create the Gaussian pyramid of the input LR image g, the same operators are applied on g, thus $\mathcal{D}_z\mathcal{B}g$ is the image at the lower scale of the Gaussian pyramid. Specifically, the blur operator $\mathcal{B}$ convolves the image with the Gaussian kernel of size f_z and standard deviation σ_z. The downsampling operator $\mathcal{D}_z$ keeps every z^{th} sample, i.e., if $p = N_1 b + a$ is a pixel position in the image g, then $\tilde{p} = \lfloor N_1/z \rfloor \lfloor b/z \rfloor + \lfloor a/z \rfloor$ is its corresponding pixel position in the image $\mathcal{D}_z\mathcal{B}g$. This downsampling scheme is illustrated in Fig. 3.10. Since in SR we are working at different scales, we can also denote a mask Ψ at a larger scale as $z\Psi$. When the scaling factor z is used for increasing the scale, we will refer to it as the magnification factor.

3.4 Single-image patch-based SR using MRF modelling

In this section, we propose a novel single-image patch-based SR algorithm. The main idea is to reconstruct the HR image patch-by-patch by using the MRF model to choose one of the candidate HR patches for each position in the HR image so that the global agreement of HR patches is enforced. Candidate HR patches come from the input LR image itself, hence the single-image approach. In this way, we reduce the set of candidate HR patches from all possible 256^{N_p} patches, where N_p is the number of pixels in a patch, to hundreds of thousands (depending on the size of the input image).

Our single-image patch-based SR method can be divided into three main phases: learning, reconstruction and post-processing (see Fig. 3.11). In the *learning* phase, we additionally constrain the set of candidate HR patches for each position in the HR image separately, by using its corresponding LR patch from the input LR image, and by exploiting patch redundancy across its different resolution scales. This patch redundancy was statistically justified in [Glasner 09, Zontak 11].

The subsequent *reconstruction* phase models the HR image as an MRF and performs inference on this model. As a result of inference, one of the candidate HR patches is chosen for each position in the HR image, so that all the HR patches agree with each other globally. The MRF model has a great advantage over the simpler alternative, i.e., choosing the best match at each location, as we will demonstrate shortly.

In the last phase of our algorithm, we apply *post-processing* techniques to eliminate remaining artefacts. We use IBP [Irani 91] to ensure the consistency of the HR result with the input LR image (see also Eq. (3.12). In the case of a small input image and a high magnification factor, the level of patch redundancy may be insufficient, which results in visible artefacts. For that reason, we also use steering kernel regression [Takeda 07] that produces a smooth and artefact-free image, while still preserving edges, ridges and blobs. The effects of the post-processing methods is further discussed in Section 3.5 and Fig. 3.14. Post-processing together with MRF modelling allows us to obtain a competitive SR result even with only having the LR image as the algorithm's input.

In order to achieve this, we combine the learning phase of [Glasner 09], by searching for candidate HR patches within the Gaussian pyramid of the input image itself, and the reconstruction phase of [Freeman 00], which uses the MRF model to reconstruct the HR image. The main benefit of this learning approach is that no external database, as a limited set of candidate HR patches, is required, which results in a faster search and absence of the "hallucination" effect, when compared to [Freeman 00]. We add another contribution to the inference part of the MRF by using a simpler and faster method instead of the slow LBP [Yedidia 01b] used originally in [Freeman 00]. We use our neighbourhood-consensus message passing (NCMP) from Section 2.4.

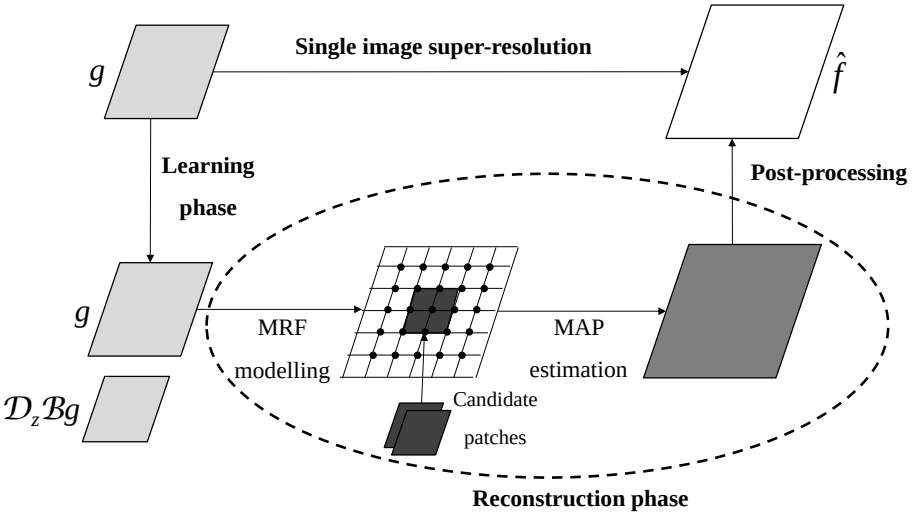


Figure 3.11: The proposed single-image patch-based SR method.

Using MRF in the reconstruction of the HR image enables us to stay in the patch-based domain without combining it with classical (mutli-frame) SR as in [Glasner 09], thus avoiding all the limitations of the multi-frame approach (discussed previously in Section 3.2.5.1).

In the remaining of this section, we will describe in details the learning and the reconstruction phase.

3.4.1 Learning candidate patches

In order to learn candidate patches, we use the example-based part of the algorithm from [Glasner 09], in the sense that we search for similar LR patches within the Gaussian pyramid of the input LR image, and use their "parent" HR patches for further reconstruction. Therefore, we will first introduce the algorithm of [Glasner 09] and then explain how our approach differs.

The goal of the learning phase, illustrated in Fig. 3.12, is to find a small set of candidate HR patches within the input image g itself for each position in the HR image f. These positions correspond to each pixel in g. For each LR patch from the input LR image g (centred at each pixel in g, excluding pixels at the border), we find the L most similar LR patches in the image $\mathcal{D}_z\mathcal{B}g$ by minimizing the SSD (step 2 in Fig. 3.12). This search exploits the patch redundancy across different scales of the Gaussian pyramid. The shape of the LR patch is determined by the mask Ψ. For each of the L most similar LR patches, we can find its "parent" patch in the input LR image (step 3 in Fig. 3.12). This "parent" patch is positioned at the corresponding pixel position and its shape is determined by the scaled mask $z\Psi$. The found matching LR patch and its "parent" patch together form an LR/HR pair, which

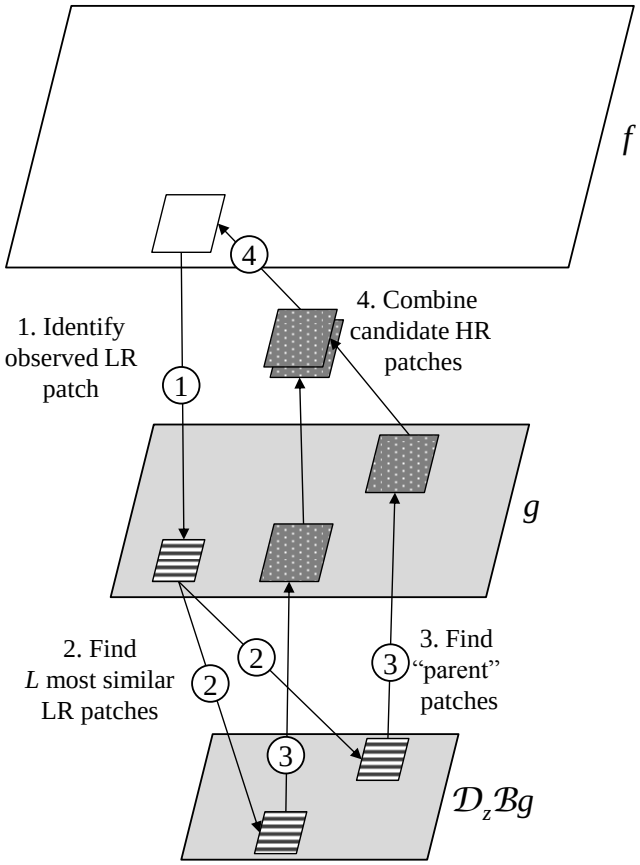


Figure 3.12: Illustration of the process of learning candidate patches.

behaves in the same way as the starting LR patch from the input LR image and its corresponding unknown HR patch. As a consequence, we can use "parent" patches as candidate HR patches for corresponding locations in the HR image. Note that this is only possible if we use the same scaling factor z for downsampling (i.e., to obtain the Gaussian pyramid) as for magnification (i.e., to obtain the HR image from the input LR image).

In [Glasner 09], multiple matching patches are chosen for each patch in the input LR image g, but the search is performed inside multiple levels of the Gaussian pyramid with a non-integer scaling factor, namely $\alpha^l = 1.25^l$, where $l = -1, \ldots, -m$. The corresponding "parent" patches of these matches represent "learned" HR patches for higher levels f_l to be reconstructed, where $l = 1, \ldots, N_l - 1$, and $z = \alpha^{N_l}$ is the desired magnification factor. To achieve the desired resolution, all these learned patches are combined in the classical multi-frame approach (see Section 3.2.4). In this approach, each of the pixels in each of the LR images induces one linear constraint on the pixel values in

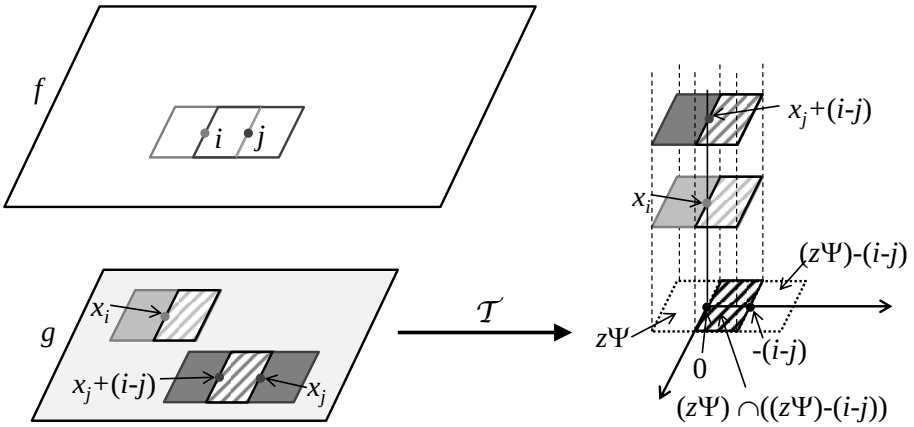


Figure 3.13: Illustration of SSD computation between labels of neighbouring nodes i and j in their region of overlap (see Eq. (3.16)).

the HR image, thus if sufficient number of LR images is available (at sub-pixel shifts), the system of linear equations becomes determined. In [Glasner 09], example-based and classical SR are combined, in the sense that each of the "learned" HR patches, as a small image, induces linear constraints on the HR image, in the form of the classical SR constraints, but with the smaller blur kernel. In order to recover the missing high frequencies, sub-pixel shifts have to exist between the learned patches, which is achieved by using a non-integer scaling factor. Furthermore, the authors of [Glasner 09] do not solve the system of linear equations at once for the desired HR image f_{N_l}, but rather they employ a coarse-to-fine strategy, i.e., different resolution levels from f_1 to f_{N_l} are reconstructed one after another, in order to achieve numerical stability. Due to the non-integer scaling factor, searching at multiple levels of the Gaussian pyramid and the coarse-to-fine reconstruction, this approach is very complex, although it delivers very promising results.

The main difference of our approach from [Glasner 09] is in the reconstruction step (step 4 in Fig. 3.12). While in [Glasner 09] the classical multi-frame approach is applied for HR image reconstruction as explained above, we use MRF modelling of the HR image (Section 3.4.2). This allows us to perform a more efficient learning phase. Specifically, we search for similar patches in only one level of the Gaussian pyramid, where the downsampling factor is equal to the magnification factor, as explained above. This is the case for magnification factor of 2 or 3, while for higher magnification factors, which must be 2 or 3 to the power of N_l ($z = \alpha^{N_l}$, where, e.g., $z = 4$, $\alpha = 2$ and $N_l = 2$), the proposed algorithm (both the learning and reconstruction phase) is applied N_l times recursively using the previously obtained SR result as an input image in a coarse-to-fine manner. However, each time the search is performed in only one level of the pyramid.

In Fig. 3.10, we can see that the downsampling operator downsamples

the image by keeping the top left pixel in each $z \times z$ block of pixels in the image at higher scale. However, there are z^2 possible options for which pixel to keep. Starting from the input image g, we generate z^2 images $\mathcal{D}_z\mathcal{B}g$ corresponding to all possible options, and use all of them as the search space. In this way, all possible patches from g are considered as parent patches. This step increases the search space and, therefore, computation time, because we search in z^2 images instead of one. However, we could perform the search more efficiently by, e.g., searching in only one image and if a good match is found, the search can be continued in a small window around the corresponding locations in all other images.[6]

The main advantages of the proposed approach in comparison with [Glasner 09] are the following. First, we can use only one level of the pyramid as the search space, whose downsampling factor corresponds to the magnification factor, instead of using multiple levels with non-integer downsampling factor, thus simplifying the search. Second, the method in [Glasner 09] reconstructs all images at intermediate resolutions (between the input and the desired resolution) in a coarse-to-fine manner, regardless of the magnification factor. We use coarse-to-fine reconstruction by applying the proposed method recursively on images at intermediate resolutions, but only for magnification factors higher than 3, and even then there are fewer intermediate levels because we use an integer downsampling factor. For magnifications factors lower than 3, we apply the proposed method only once to obtain the HR image at the desired resolution. Finally, we avoid sub-pixel registration, which often causes inaccurate results.

3.4.2 High-resolution image reconstruction

Using the MRF framework to perform HR image reconstruction was proposed in [Freeman 00]. In this approach, an HR patch is assigned to each position in the HR image, taking into account both the agreement of the HR patch with the available data (the input image) and the agreement of neighbouring HR patches in their overlap region. Furthermore, the image is observed as a whole rather than a collection of local assumptions. In this respect, the choice of patches is formulated as a global optimization problem over the whole HR image by using the MRF framework [Li 95] (see also Chapter 2).

Motivated by [Freeman 00], we adopt such MRF framework for the reconstruction phase of our approach. We model the HR image f as an MRF (see Fig. 2.3 and Section 2.2.4 for more details), where the lattice S of MRF nodes consists of pixel positions i, which are z pixels apart in horizontal or vertical direction on the HR lattice. We consider the first-order neighbourhood system with pairwise cliques $\langle i, j \rangle$. The values that are to be assigned to nodes are the candidate HR patches. These values are usually referred to as labels in MRF theory (see also Section 2.2.1). Each MRF node i is also associated with an observation (measured data), which, in this case, is the LR patch centred

[6] This approach was not explored in the current implementation.

Figure 3.14: The cropped "castle" image with $\times 2$ magnification. From left to right and top to bottom: bicubic interpolation, MRF result, MRF with IBP, and MRF with IBP and kernel regression.

at the pixel in g, which corresponds to the position i in the HR image. Since all observations are from the same image g and they are specified by the same mask shape Ψ, we will refer to an observation by its position in the image g, denoted as y_i (see Section 2.2.4).

As we discussed at the beginning of this section, considering all the possible HR patches as labels would be prohibitive, thus it is necessary to limit the number of labels in some meaningful way. We achieve this by using the learning phase described in Section 3.4.1. Specifically, L labels for each node i are the L "parent" patches from the input LR image g (the dotted patches in Fig. 3.12), whose shape is determined by the mask $z\Psi$. They correspond to the L most similar LR patches of the observation (LR patch) at the node i. Like for observations, we refer to labels by their position in the image g. Then the

label *position* set Λ_i of the node i consists of the positions of the found "parent" patches. The assignment of a label to the node i amounts to copying the values from the patch positioned at $x_i \in \Lambda_i$ to the positions within the mask $z\Psi + i$ in the HR image.[7] Note that the masks centred at neighbouring nodes are overlapping, unless the square mask Ψ consists of a single pixel (which is never the case).

To completely define the model, we still have to define compatibility functions between an observation and a label at each node (the so-called local evidence) and between labels of neighbouring nodes. As a reminder, the former determines how much a label agrees with measured data, and the latter encodes prior information on the distribution of the unknown image. As in [Freeman 00], the local evidence is given as the Gaussian function of the matching error between the observed LR patch centred at y_i in g and its L most similar LR patches in the image $\mathcal{D}_z\mathcal{B}g$:

$$\phi(x_i, y_i) = \exp\big(- \|\mathcal{D}_z\mathcal{B}\mathcal{T}_{x_i}g - \mathcal{T}_{y_i}g\|^2_\Psi / 2\sigma^2_{loc}\big), \tag{3.15}$$

where the operators and the norm are defined in Section 3.3. Note that the positions of the L most similar LR patches are related to the label positions x_i via the downsampling operator $\mathcal{D}_z$, because the labels are their "parent" patches. The compatibility between the labels of neighbouring nodes is the Gaussian function of the matching error between these labels in their nodes' region of overlap (see Fig. 3.13 for graphical representation):

$$\psi(x_i, x_j) = \exp\big(- \|\mathcal{T}_{x_i}g - \mathcal{T}_{x_j+(i-j)}g\|^2_{(z\Psi)\cap((z\Psi)-(i-j))} / 2\sigma^2_{com}\big). \tag{3.16}$$

σ_{loc} and σ_{com} are the noise standard deviations, which represent the difference between some "ideal" training samples and our image, and training samples, respectively.

Now, we have to choose one label at each node that fits the above constraints the best over the whole graph. This can be achieved by finding the MAP estimate as

$$\begin{aligned} \hat{\mathbf{x}}_{MAP} &= \arg\max_{\mathbf{x}} P(\mathbf{x}|\mathbf{y}) \\ P(\mathbf{x}|\mathbf{y}) &\propto \prod_{\langle i,j\rangle} \psi(x_i, x_j) \prod_i \phi(x_i, y_i), \end{aligned} \tag{3.17}$$

where $\phi(x_i, y_i)$ is defined in Eq. (3.15) and $\psi(x_i, x_j)$ in Eq. (3.16). This is generally a difficult problem to be solved exactly, but there is a number of approximate inference algorithms that can yield an approximate solution (see

[7] In Chapter 2, x_i and y_i denoted the label and the observation themselves, respectively, and Λ_i denoted the label set. For the sake of compactness of representation, in this chapter and Chapter 5, we use x_i and y_i to denote the label and the observation *position*, respectively, and Λ_i to denote the label position set.

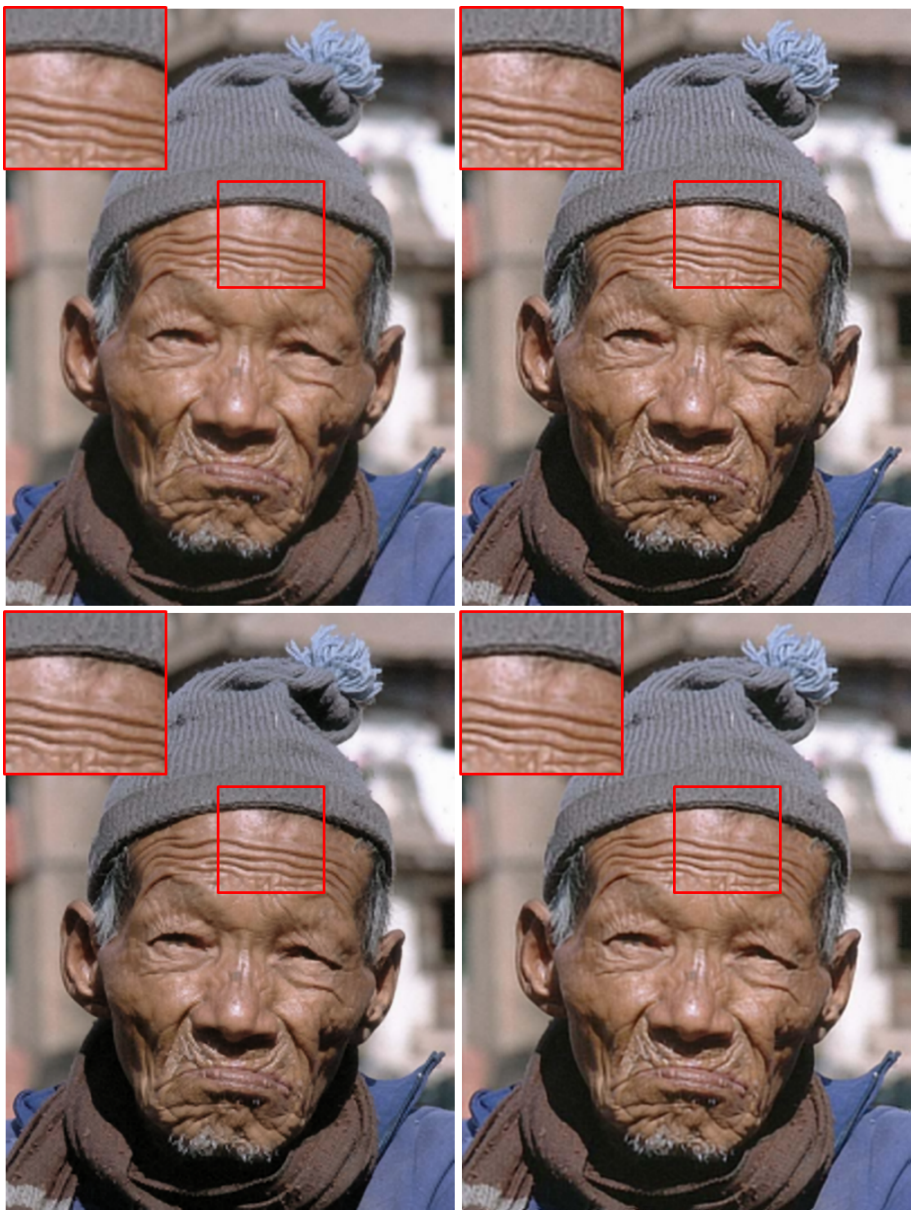


Figure 3.15: The "old man" image: results of image upscaling methods for $\times 2$ magnification from the input image of size 200×266. From left to right and top to bottom: bicubic interpolation, IBP [Irani 91], Fattal [Fattal 07], and the proposed method.

Section 2.3 for an overview). The method from [Freeman 00] uses LBP as an inference method (see Section 2.3.5). We use our NCMP inference method

(Section 2.4), which is simpler and faster than LBP. After the specified number of iterations of NCMP, at each node i one label positioned at $\hat{x}_i$ is chosen. We then obtain the final SR result as $\hat{f}(p+i) = g(p+\hat{x}_i)$, $\forall p \in z\Psi$, $\forall i \in S$, where the values in the overlap region between the neighbouring nodes are averaged.

Although we use the MRF model from [Freeman 00] in this reconstruction phase, our approach has several major differences. First of all, our candidate HR patches, i.e., labels, are obtained from the input image itself, while in [Freeman 00] an external database is used.[8] Moreover, they consist of raw pixel values instead of high-frequency details (see Section 3.2.5.1 and Fig. 3.8). Therefore, our method does not require pre-processing of the search space and the input image. Finally, we use our inference method for optimization, introduced previously in Section 2.4.

Instead of using the MRF approach for HR image reconstruction, we could simply choose the best match for each LR patch and use the "parent" patch of that best match as the HR patch for the corresponding position in the HR image. Again, we can take the average in the overlap region between the neighbouring HR patches. Although this solution could speed up the search process (because we only search for one match), the resulting image will have visible artefacts, as shown in the top right of Fig. 2.14. In the bottom of Fig. 2.14, we can see that using an MRF model produces much better result, even if we use a simple inference method like NCMP (bottom right image).

3.5 Results

We applied the proposed method to enhance the resolution of natural images of different sizes. In particular, we applied the proposed method on the luminance channel of the image, while chrominance channels were upscaled with bicubic interpolation. The size of the input image determines the size of the search space, i.e., a bigger input image provides a bigger search space. This means that there is more chance of finding a good match in the learning phase and thus, the final result can be better. We demonstrate the effectiveness of our technique for a sufficiently large search space in the first experiment. Fig. 3.14 shows the cropped part of the "castle" image (481×321 pixels) and the results of our SR algorithm with the magnification $z = 2$. It can be seen in the top right that the output of the MRF, without any post-processing, gives already reasonably good results. For example, all edges are sharp without jaggy artefacts, which are visible in the result of bicubic interpolation (top left). Fig. 3.14 also demonstrates the effect of the post-processing methods. We can see in the bottom left that back-projection further improves the MRF result by eliminating artefacts and enhancing textures (e.g., texture on the roof). Finally, kernel regression (bottom right of Fig. 3.14) only slightly smooths the image, and can even be left out as a post-processing step in this case. The amount of post-processing needs to be adjusted in order to avoid introducing additional

[8] In [Freeman 00], using the input image as a source of labels was tested, but not explored in detail.

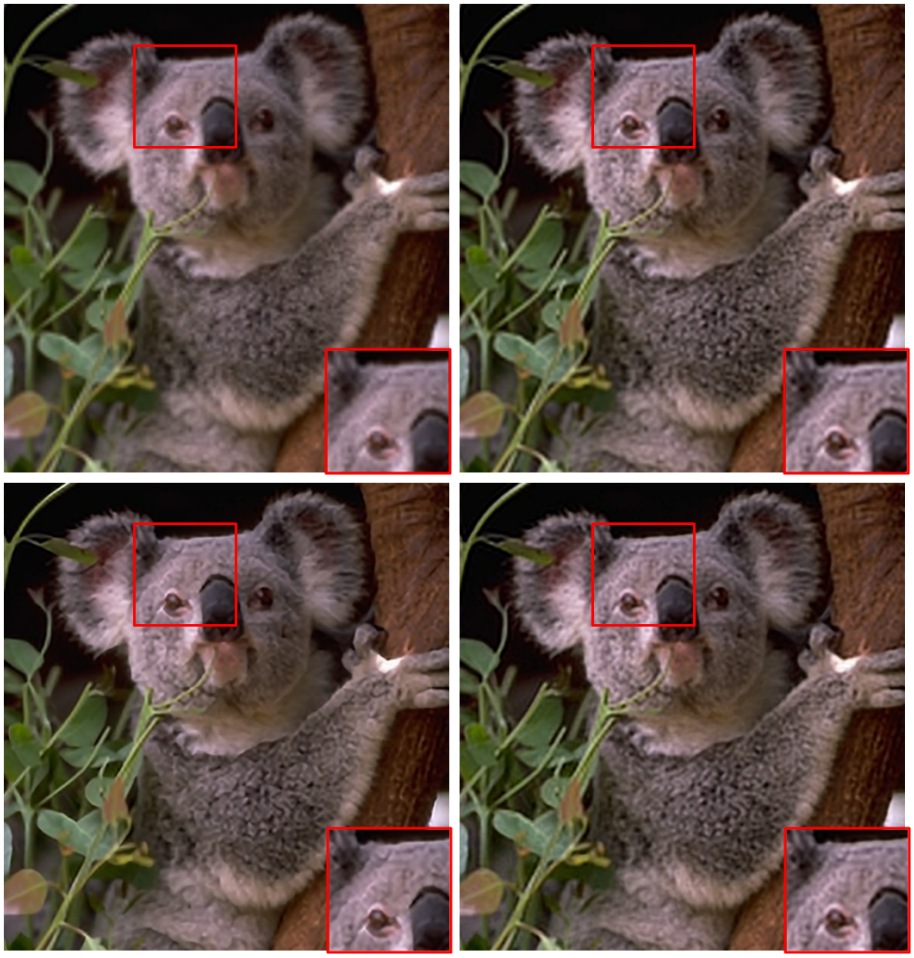


Figure 3.16: Cropped version of the "koala" image: results of image upscaling methods for $\times 3$ magnification from the input image of size 161×241. From left to right and top to bottom: bicubic interpolation, IBP [Irani 91], Glasner et al. [Glasner 09], and the proposed method.

artefacts. This means properly choosing the number of iterations and the kernel parameters in order to avoid ghosting artefacts (for back-projection) or over-smoothing (for kernel regression).

In the next experiment, we compare the proposed method with the standard bicubic interpolation, IBP [Irani 91], the original example-based technique from [Freeman 00][9], and the state-of-the-art techniques from [HaCo-

[9] http://people.csail.mit.edu/billf/

Figure 3.17: Cropped version of the "sunflowers" image: results of image upscaling methods for $\times 3$ magnification from the input image of size 320×208. From left to right and top to bottom: bicubic interpolation, IBP [Irani 91], Glasner et al. [Glasner 09], and the proposed method.

Figure 3.18: The "pumpkins" image: results of image upscaling methods for ×4 magnification from the input image of size 160×128. From left to right and top to bottom: bicubic interpolation, IBP [Irani 91], Freeman et al. [Freeman 00], Fattal [Fattal 07], HaCohen et al. [HaCohen 10], and the proposed method.

hen 10][10] and [Glasner 09][11], the latter being another single-image SR method, and [Fattal 07][12], which is a reconstruction-based method (Section 3.2.3). The

[10] http://www.cs.huji.ac.il/∼yoavhacohen/upsampling/
[11] http://www.wisdom.weizmann.ac.il/∼vision/SingleImageSR.html
[12] http://www.cs.huji.ac.il/∼raananf/projects/upsampling/upsampling.html

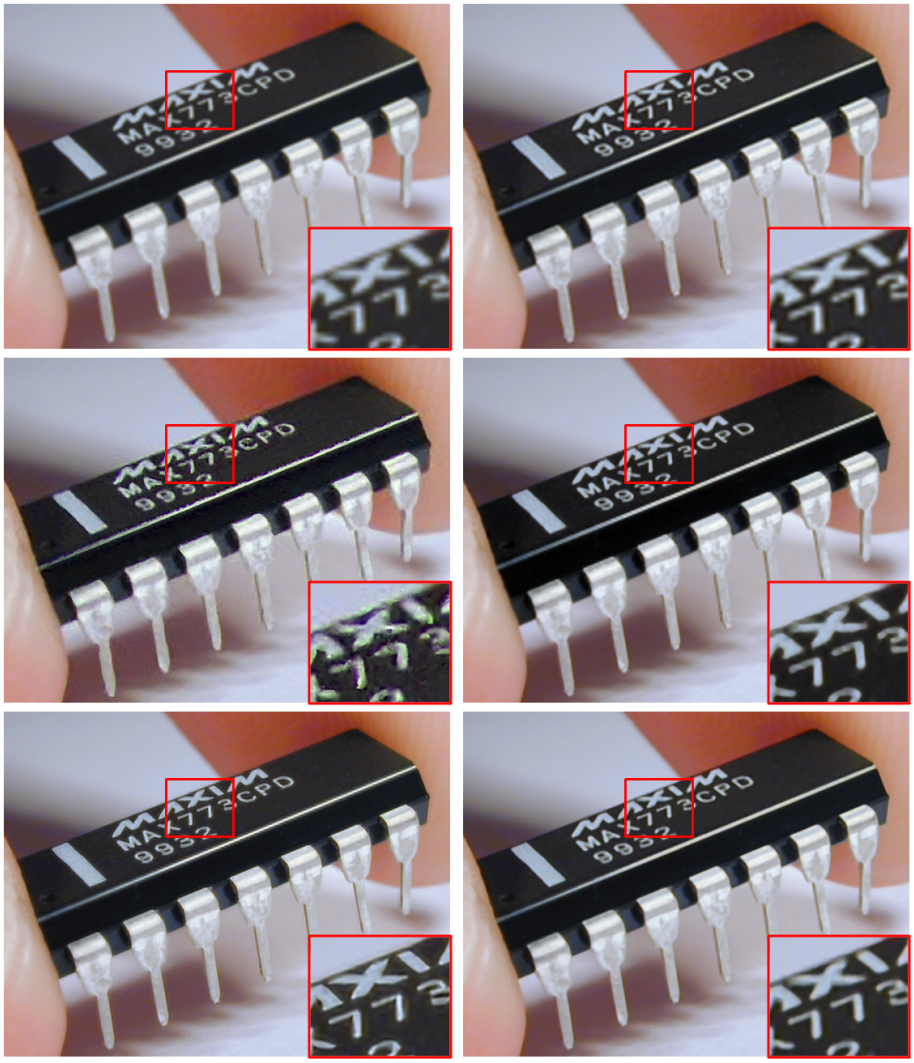


Figure 3.19: Cropped version of the "chip" image: results of image upscaling methods for $\times 4$ magnification from the input image of size 244×200. From left to right and top to bottom: bicubic interpolation, IBP [Irani 91], Freeman et al. [Freeman 00], Fattal [Fattal 07], Glasner et al. [Glasner 09], and the proposed method.

input images and the results of the reference methods were downloaded from their websites, except for the results of [Freeman 00], which were generated using the software available on the author's website. For the proposed method, we chose the following parameters: $\sigma_{loc} = 0.1$ and $\sigma_{com} = 0.5$ for MRF compatibility functions, the size of the Gaussian kernel $f_z = 3$ (for magnification factor $z = 2$) and $f_z = 5$ (for $z = 3$). These parameters were chosen so to

Figure 3.20: Images used for quantitative comparisons shown in Tables 3.1 and 3.2. All used images are of the same size, but in this figure they are differently scaled for illustrative purpose.

Figure 3.21: Cropped version of the "bears" image (x4 magnification). From left to right and top to bottom: original image, result of Freeman et al. [Freeman 00], bicubic interpolation, IBP [Irani 91], and the proposed method.

produce the best results (with the fewest artefacts) for different images. The standard deviation was chosen as $\sigma_z = z$, as suggested in [Glasner 09]. We set the LR patch size to 3×3 (resulting in HR patch size $3z \times 3z$), because with other choices of patch size we were not able to find sufficient number of similar patches due to the limited size of our search space. The choice of the number of similar patches $L = 10$ was motivated by the statistical analysis in [Glasner 09], where it was concluded that about 80% of image patches have 9 or more similar patches in the lower level of the Gaussian pyramid with the downsampling factor z between 2 and 3.

The results of the proposed method and their comparisons with the previously mentioned reference methods are shown in Figs. 3.15 - 3.19 for different values of the magnification factor: $z = 2$ (Fig. 3.15), $z = 3$ (Figs. 3.16 and 3.17) and $z = 4$ (Figs. 3.18 and 3.19). In Fig. 3.15, the result of the proposed method is somewhat sharper with fewer jaggy artefacts compared to the results of the reference methods, although the difference is not so significant since magnification factor is rather small. For $z = 3$ (Figs. 3.16 and 3.17), the

Table 3.1: RMSE comparison of the results of our method and the reference methods.

Image	Bicubic	IBP	Freeman et al.	Our
"Butterfly" x2	0.1486	0.1357	0.1630	**0.1252**
"Skyscraper" x2	0.2794	0.2584	0.3054	**0.2441**
"Zebras" x2	0.3951	0.3546	0.4410	**0.3425**
"Bears" x4	0.3085	0.3033	0.3481	**0.3024**
"Ladybirds" x4	0.1730	0.1688	0.1969	**0.1582**
"Church" x4	0.1775	0.1649	0.1844	**0.1366**

Table 3.2: SSIM comparison of the results of our method and the reference methods.

Image	Bicubic	IBP	Freeman et al.	Our
"Butterfly" x2	0.9565	0.9637	0.9379	**0.9677**
"Skyscraper" x2	0.9160	0.9358	0.8898	**0.9396**
"Zebras" x2	0.9044	**0.9340**	0.8633	0.9269
"Bears" x4	0.7061	0.7352	0.6612	**0.7385**
"Ladybirds" x4	0.9287	0.9311	0.9040	**0.9365**
"Church" x4	0.8962	0.8996	0.8877	**0.9311**

proposed method and the method from [Glasner 09] are able to produce sharp edges without jaggy artefacts present in the results of bicubic interpolation and IBP. Additionally, the proposed method reconstructs the textured area (e.g., fur of the koala) somewhat better than [Glasner 09]. For $z = 4$ (Figs. 3.18 and 3.19), we can see that in the result of [Freeman 00], many additional artefacts appear because of using examples from an external database. The same happens in the result of [HaCohen 10] in Fig. 3.18, although much less because the database is not general, but specifically selected by the user based on the image content. The result of [HaCohen 10] is the sharpest for this image, but the texture appears quite unrealistic. The results of bicubic interpolation, IBP and the method from [Fattal 07] on both images suffer from blurry and jaggy edges, while in Fig. 3.19 method from [Glasner 09] produces ringing artefacts. On the other hand, our method gives sharp and realistic result without any additional artefacts.

In Tables 3.1 and 3.2, we give quantitative results for a few images from Berkeley segmentation database[13] shown in Fig. 3.20. We performed this experiment by first downsampling the original images by factor 2 or 4, and then we used the proposed SR method and the reference methods, namely bicubic interpolation, IBP and the example-based method from [Freeman 00], to bring the images back to the original resolution. We calculated the root mean square error (RMSE), shown in Table 3.1, and the structure similarity index (SSIM) [Wang 04], shown in Table 3.2, between the image upscaling results

[13] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench

and the ground truth on grey-scale versions normalized to zero mean and unit variance. It can be seen that our method produces for all images the smallest RMSE (the lower the better), and for all images except the "zebras" image the highest structure similarity score (the higher the better). The quantitative improvement is, however, limited, since the improvement is concentrated in edge regions, which represent a small portion of the whole image. Note also that the method from [Freeman 00] gives the worst quantitative results due to artefacts introduced by using an external database. We demonstrate the qualitative comparison for a small part of the "bears" image in Fig. 3.21. We can see that the proposed method gives the sharpest result, without jaggy or ringing artefacts present in the result of IBP, although this advantage is not so evident from the quantitative comparison. However, none of the methods is able to reconstruct texture (fur of the bear or waves in the water) because the input image is too small, only 120×80, and almost all the texture was lost in the degradation process.

3.6 Conclusion

In this chapter, we introduced the image upscaling problem through basic definitions and observation models, which relate HR and LR image in the case of image upscaling from a single input LR image, and HR image and multiple LR images in the case of the multi-frame approach. We also defined the image upscaling problem from a regularization point of view, since those concepts form a basis for some of the image upscaling methods. Furthermore, a categorization and systematic overview of different image upscaling methods were given, with special emphasis on example-based (patch-based) methods.

The main contribution of this chapter is a novel single-image SR method based on MRF modelling, making use of concepts and tools studied in Chapter 2. In the proposed method, unknown HR image was modelled as an MRF, where the unknown image attributes to be estimated at MRF nodes are overlapping HR patches. Possible candidate HR patches (i.e., labels) for these nodes are found within only one level of the Gaussian pyramid of the input LR image. To choose the best candidate in the MAP sense, we used our previously developed NCMP inference method (described in Section 2.4), which makes this step fast and simple. Additionally, we performed IBP and steering kernel regression to further improve the results. Visual and quantitative comparison of results (in terms of RMSE and SSIM) shows that our method greatly outperforms standard techniques, while being visually better or comparable with state-of-the-art techniques. This work resulted in one conference publication [Ružić 11b].

The proposed method was developed as a general-purpose SR algorithm, without making assumptions about camera parameters and the type of the scene captured by the camera. It was evaluated on natural images, with resolution ranging from 120×80 to 481×321. Nowadays, there is also the need to convert images of high resolution to an even higher resolution (e.g., HD to

4K). Our method could also be used to perform such tasks, with the advantage of having bigger search space at disposal, but at the higher computation cost. Moreover, it could be used as a pre-processing step for certain computer vision tasks, e.g., face and licence plate recognition, optical character recognition of text documents, etc. (see also Section 3.1.1). However, in order for the results to be competitive with SR algorithms that are especially developed for these types of problems, we should incorporate specifics of these problems into the algorithm, e.g., in the form of prior knowledge. Finally, the above mentioned computer vision tasks could also be used to evaluate and compare the results of different methods, i.e., one could evaluate how much an SR algorithm improves feature detection or recognition. This type of evaluation was not performed within this research, but it represents a possible direction for future research.

The potentials of patch-based approaches that we witnessed while exploring SR application gave us the motivation to further pursue patch-based methods in other applications.

4

Context-aware patch-based image inpainting

Digital image inpainting, or image completion, is an image processing task of filling in the missing or damaged region in a digital image in a visually plausible way. In order to solve this problem, usually the available data, which lies outside the region to be inpainted, is used. Image inpainting has become an active research field in image processing due to its broad area of applications in image and video restoration and editing.

We begin this chapter by introducing the problem of image inpainting in Section 4.1. Next, we give an overview of inpainting methods, which can be categorized into geometry-based (Section 4.2) and patch-based (Section 4.3). Patch-based methods, being the focus of this thesis, are visited more in-depth, and we classify them according to the type of patch selection, patch search and patch priority they employ. Two most important contributions of this chapter are: 1) a novel strategy for context-aware patch selection (Section 4.4), which can be used with any patch-based inpainting method, and 2) a novel inpainting method (Section 4.5), which uses the proposed context-aware approach. The idea of the proposed approach is to employ contextual (textural) descriptors of image regions to guide and improve the inpainting process.

4.1 Introduction

The term *inpainting* is related to artwork restoration, which dates all the way back to renaissance times, when medieval artwork was retouched or inpainted in order to bring the painting "up-to-date" or to fill in the gaps [Walden 85]. This practice naturally extended from paintings to photography and film, with the purpose to, e.g., remove cracks, dust and scratches or add/remove elements or objects by "airbrushing" [King 97].

This kind of inpainting refers to *physical* altering of a painting or a photograph. Nowadays, image inpainting is performed digitally on digital images. Digital image inpainting (or simply image inpainting) covers a wide

range of applications, such as

- image restoration (e.g., scratch removal),
- photo-editing (e.g., object or text removal),
- recovery of missing blocks in image and video transmission,
- virtual restoration of digitized paintings (crack removal),
- virtual view synthesis for 3D video (disocclusion), etc.

Image inpainting is related to texture synthesis [Efros 99], in the sense that texture synthesis also introduces some new content to the image that was not present before. However, image inpainting is a more demanding task due to the variety of textures that are present in the known (undamaged) part of the image and that need to be replicated inside the missing region. Furthermore, there are also linear structures, i.e., object contours and borders, that need to be considered. Therefore, texture synthesis is typically just one part of the image inpainting technique.

Some image inpainting methods also incorporate automatic detection of damaged regions, e.g., scratches in film [Kokaram 95b] and cracks in digitized paintings [Cornelis 13]. However, the vast majority of techniques consider the missing or damaged region to be known beforehand, either through user interaction or above mentioned automatic detection.

Despite the huge variety of inpainting methods, they are all based on the same methodology, which is derived from the way conservators perform physical inpainting on actual paintings. The methodology of conservators is the following [Bertalmio 00]:

1. The content of the whole image dictates how to fill in the missing region (the "hole"), since the goal of inpainting is to restore the unity of the work.
2. The structure of the area surrounding the missing region is continued inside the hole, by drawing the contour lines as prolongation of those arriving at the border of the hole.
3. Different areas inside the missing region, defined by the continued contour lines, are filled with colour in accordance with the colour at the border of the missing region.
4. The small details are painted, i.e., the texture is added.

Different digital inpainting methods attempt to perform some or all of these steps in order to yield a visually plausible inpainted image. Among many different image inpainting techniques, two categories can be distinguished: *geometry-based* and *patch-based*. Big part of this chapter contains the overview of these methods, with the emphasis on patch-based methods. The exact location and the extent of the missing region is assumed to be known (or determined by another method).

4.2 Geometry-based methods

Geometry-based methods, e.g., [Bertalmio 00, Shen 03, Ballester 01, Tschumperlé 06], fill in the missing region by smoothly propagating image content from the boundary to the interior of the missing region. The focus of these methods is on propagating linear structures by continuing lines that arrive at the border of the missing region inside the hole (step 2 of the methodology from Section 4.1). Hence the name *geometry-based inpainting* [Bertalmio 06]. In literature, this group of methods is most often referred to as diffusion- or partial differential equations (PDE)-based. We find that these terms do not describe well this wide group of methods and that they represent just its sub-categories, as will be explained shortly.

Geometric inpainting stems from the psychophysical analysis of human vision [Kanizsa 79], where it was noted that continuation of object boundaries plays a crucial role in the process of missing region recovery (the so-called "amodal completion"). The continuation process should be as smooth as possible, i.e., smooth completion curves are preferred to abrupt changes of direction. Based on this observation, Nitzeberg et al. proposed a disocclusion algorithm for the purpose of image segmentation and depth estimation in [Nitzberg 93]. Their idea was to detect edges and T-junctions (points where edges form a "T"), and then connect T-junctions at approximately the same grey level with a new edge of minimum length and curvature. This was achieved by a variational process, in particular by minimizing an energy functional called Euler's elastica [Mumford 94]. However, the algorithm was not suitable for natural images and it was based on the detection of edges, which is insufficiently reliable and precise. Nevertheless, this approach is considered to be a pioneering work in the recovery of plane image geometry, from which two categories of geometry-based inpainting algorithms evolved: *variational* and *PDE-based* approaches.[1]

Variational approaches are related to the Bayesian framework, in the sense that the way of inpainting an image depends on the existing part of the image (the *data* model) and prior knowledge about the type of the image (described by the *prior* model). The choice of the prior model is crucial for the success of the algorithm. The prior model should be general, i.e., independent of a specific object class, but it should comply with geometric regularities of the object [Shen 03]. Some models that were used for variational inpainting are total variation (TV) model [Rudin 92,Shen 02], joint vector-field and grey-value model [Ballester 01], Mumford-Shah(-Euler) model [Mumford 89,Esedoglu 02], and already mentioned Euler's elastica model [Mumford 94, Masnou 98, Masnou 02, Shen 03].

The work of Masnou and Morel [Masnou 98,Masnou 02] was the first variational approach for inpainting, which extended the ideas from [Nitzberg 93] to disocclusion in natural images. They proposed to use level (isophote) lines, i.e., the lines of equal grey values, because they give a reliable, complete and contrast-invariant representation of an image [Masnou 98]. The idea is to fill

[1] For this reason, we restrain from naming the whole group of methods PDE-based.

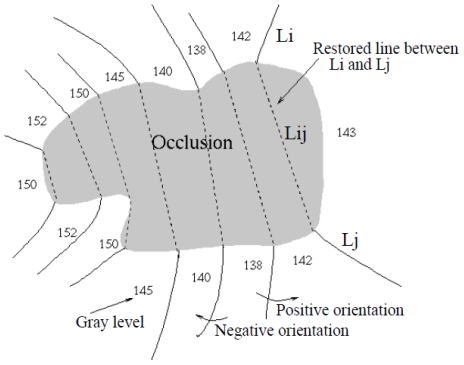


Figure 4.1: Illustration of possible connections between pairs of isophotes inside the occlusion (figure taken from [Masnou 98]).

in the missing region by joining with geodesic curves the points of isophotes arriving at the missing region's boundary (see Fig. 4.1). However, the model could only recover straight lines and its practical implementation via dynamic programming was limited to the application on images with a specific topology of the missing region. In particular, the missing region cannot contain a hole and its boundary cannot contain any self-crossings.

Other variational approaches do not have the limitation to specific topology due to the numerical PDEs used to minimize the energy functional. They differ in complexity, contrast-invariance, the ability to recover curvy lines (the so-called *principle of good continuation* [Bertalmio 06], achieved by equations of at least third-order [Bertalmio 01]), and the ability to connect widely separated parts of one object (the so-called *connectivity principle* [Shen 03]). For example, the method in [Shen 03] can recover curvy level lines and is contrast-invariant, but due to the high order (fourth) of the model, the stability and convergence speed can become an issue. The method in [Shen 02] and Mumford-Shah model in [Esedoglu 02] are simple (second-order), but they can only recover straight or polygonal lines and they are unable to fill in bigger missing regions. The Mumford-Shah-Euler model in [Esedoglu 02] improves on this behaviour at the expense of higher complexity. The method in [Ballester 01] seems to be the most promising, in the sense of reconciling the demands for low complexity and the ability to recover arbitrarily-shaped lines regardless of the size of the missing region (to some respect, see [Bertalmio 06]). However, it is not contrast-invariant, although this characteristic is imposed in the implementation.

PDE-based methods directly introduce PDEs to perform inpainting, but, unlike in variational methods, these PDEs do not minimize any known functional. The first PDE method was introduced in [Bertalmio 00], which uses edge propagation to smoothly extend isophotes arriving at the border of the missing region, thus it involves a *propagation* process. In [Bertalmio 01], the connection between this PDE and classical fluid dynamics was shown. Further-

more, in [Bertalmio 00,Chan 01a], it was recognized that, although propagation process allows grey-scale information to propagate inside the missing region, still the *diffusion* process is necessary to stabilize the propagation and regularize the geometry of the isophotes.[2] In [Bertalmio 00], this was achieved with intermediate steps of anisotropic diffusion [Perona 90]. Some PDE-based methods employ only the diffusion process, e.g., [Chan 01b, Tschumperlé 00, Tschumperlé 06], while some attempt to model the combination of propagation and diffusion processes, e.g., [Bertalmio 01, Chan 01a]. On the other hand, the method in [Bertalmio 06] views the inpainting problem as a particular case of image interpolation and derives the third-order PDE. Like in variational approaches, the success of these methods is measured based on their contrast-invariance [Chan 01a, Chan 01b, Bertalmio 06], principle of good continuation [Bertalmio 00, Bertalmio 01, Bertalmio 06], and connectivity principle [Chan 01b, Bertalmio 06].

Geometric inpainting yields good results when inpainting long thin regions, e.g., in the application of scratch and text removal. However, they fail to replicate big textured areas and, in general, introduce blur when filling in larger holes. The inability of texture replication is due to the application of PDEs, which model a geometric process and thus allow the recovery of object contours using geometric information. On the other hand, no assumptions are made about statistics of pixel intensities, which would enable texture recovery, as in texture synthesis application [Efros 99]. Therefore, a solution for good recovery of both structure and texture in the missing region would be to combine geometric inpainting and texture synthesis, as in, e.g., [Bertalmio 03, Drori 03]. Furthermore, all the patch-based methods, which will be reviewed in the following section, also attempt to achieve this goal.

4.3 Overview of patch-based inpainting methods

Patch-based inpainting methods, in general, fill in the missing region patch-by-patch by searching for well-matching replacement patches (i.e., candidate patches) in the available part of the image and copying them to corresponding locations (see Fig. 4.2). Since candidate patches can originate from all over the image, this process is non-local. Some methods, e.g., [Jia 03, Wexler 07, Bugeau 10, He 12], deviate from this general paradigm in the sense that they recover the missing region pixel-by-pixel, but they still search for well-matching patches to find the candidate pixels.

The general idea of patch-based methods originates from patch-based texture synthesis [Efros 99, Efros 01, Wei 00, Kwatra 03]. Texture synthesis alone is, however, insufficient for satisfactory inpainting result, so patch-based inpainting methods also pay attention to structure propagation by defining the filling order [Criminisi 04, Komodakis 07, Fang 09, Xu 10, Le Meur 11], using the human intervention [Sun 05], or decomposing the image into structure and

[2] Because we consider diffusion process as one of the necessary components for geometric inpainting, we do not name the whole group of methods diffusion-based to avoid confusion.

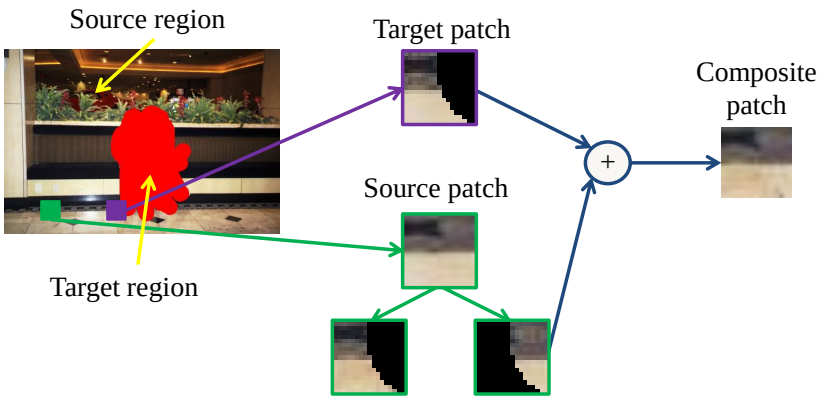


Figure 4.2: Schematic representation of the basic idea behind patch-based inpainting methods.

Figure 4.3: Comparison of geometry-based and patch-based method for inpainting big missing region. From left to right: original image with missing region marked in black, inpainting result of the geometry-based method from [Tschumperlé 06], and inpainting result of the patch-based method from [Criminisi 04].

texture components [Bertalmio 03, Drori 03, Voronin 12]. This reasoning can be summarized into two main observations of patch-based inpainting [Criminisi 04]: 1) patch-based synthesis is sufficient for both texture and structure replication, and 2) filling order of the missing region is crucial for the success of the algorithm. Compared to geometry-based methods, patch-based methods produce better results, especially when inpainting large missing regions (see Fig. 4.3 for comparison).

We identify three main components of patch-based methods:

- **Patch selection** deals with the problem of selecting candidate patches from the known region, in the sense whether to choose one or multiple matches among all the available patches and based on which information.
- **Patch search** defines how to perform the search for candidate patches across the known region. For example, a naive approach would be to search among all possible patches in the known region, although more sophisticated ways can be explored.

- **Filling order** of the missing region can be defined based on different image features.

We will classify patch-based methods according to the patch-selection process they employ in Section 4.3.1, and then we will identify sub-categories based on patch search (Section 4.3.2) and filling order (Section 4.3.3).

4.3.1 Patch selection

Based on patch selection, patch-based methods can be categorized into *"greedy"* [Drori 03, Criminisi 04, Cheng 05, Fang 09, Anupam 10, Ružić 11a, Ružić 12a], *multiple-candidate* [Wong 08,Shen 09,Xu 10,Le Meur 11,Voronin 11, Le Meur 12], and *global* [Sun 05, Komodakis 07, Wexler 07, Huang 07, Yang 09, Bugeau 10, Ružić 12b]. Greedy methods choose only one best match (called the *source* patch) for each patch to be filled (called the *target* patch), and use this source patch as such to replace the missing pixels. Multiple-candidate methods, on the other hand, choose multiple candidate patches, as the name implies, and the final patch represents either their weighted average [Wong 08, Le Meur 11, Voronin 11, Le Meur 12] or their sparse combination [Shen 09, Xu 10]. In general, for both greedy and multiple-candidate methods the matching is performed based on the known pixels of the target patch (see Fig. 4.2), within an iterative process that gradually completes the missing region. However, this matching based on the known part of the target patch can be ambiguous, in the sense that the found match will correspond to that known part, but that does not necessarily mean it is well suited for the missing part. Thus, it would be beneficial to include wider neighbourhood into the patch-selection process. Global methods attempt to achieve this by defining inpainting as a global optimization problem. They allow the choice of multiple candidates, which is here made based on the known pixels of the target patch, but also based on the neighbouring information. Eventually, one of these candidates is chosen for each position so that the whole set of patches (at all positions) minimizes a global optimization function.

The matching criterion for choosing the candidate patch (or patches) is usually the sum of squared differences (SSD) between the known pixels in the target patch and the corresponding pixels in the candidate patch, unless stated otherwise.

4.3.1.1 Greedy methods

The basic idea of greedy methods is the following: for each patch at the border of the missing region (i.e., the target patch), find *only* its best-matching patch from the known (source) region (source patch) and replace the missing pixels with the corresponding pixels from that match, until there are no more missing pixels (see Fig. 4.2). In this way, both texture and structure are replicated. Preserving structures is achieved by defining the filling order. Priority should be given to the target patches that contain object boundaries and less missing pixels.

The best known method from this group was proposed by Criminisi et al. [Criminisi 04], and it is considered as a seminal work for patch-based inpainting in general. This algorithm defines priority as a combination of the amount of reliable information within the target patch (called the *confidence* term) and the strength of isophote lines hitting the border of the missing region (called the *data* term). Furthermore, patch search is performed in an exhaustive manner, meaning that all possible patches from the source region are considered. The method in [Anupam 10] represents a fast and improved version of the algorithm in [Criminisi 04]. The improvement is achieved by performing a limited search for patches (discussed later in Section 4.3.2) and defining priority using a different combination of essentially the same features (Section 4.3.3).

The approach of [Fang 09] greatly accelerates the computation by using a multi-resolution approach, a limited patch search, a trained database of patches, and only a thin border of the patch for matching (the so-called O-shape pattern). The Euclidean distance is used as the matching criterion. Interesting property of this algorithm is the trained database, which is formed by taking O-shape patterns of all the possible candidate patches from all resolution levels of the input image (except original level), on which principal component analysis (PCA) [Jolliffe 02] is applied to reduce the dimensionality of each pattern. Then each patch is represented by a weight vector of reduced dimensionality, obtained by projecting its O-shape pattern onto this PCA eigenspace. Additionally, weight vectors are clustered using vector quantization, which speeds up the search process. During the image completion process, the input image is inpainted from low to high resolution, using the lower resolution for initialization. This method offers an alternative priority definition (discussed later in Section 4.3.3), which is additionally used to discriminate between non-directional and directional search for the best-matching patch, as an alternative to exhaustive search (Section 4.3.2). This method is greedy from the point of few of patch selection, but it introduces some interesting concepts that speed up the patch search and improve the priority definition. However, it is rather complicated and inconsistent, in the sense that almost each resolution level requires different synthesis process.

Instead of enforcing the continuation of image structures by defining the filling order, image segmentation could be used to recognize these structures as partitioning curves between the segments. These can then be extrapolated inside the missing region. Another advantage of using segmentation would be to constrain the patch search to each of the segments, depending to which segment the target patch belongs (see also step 3 of inpainting methodology in Section 4.1). One such method is proposed in [Jia 03], where tensor voting [Medioni 00] is used for both curve extrapolation and inpainting. Tensor voting is a non-iterative method that addresses the problem of structure inference from sparse data. In order to infer the value of the missing *pixel*, the texture and colour information in the neighbourhood around each pixel is described by an adaptive N-dimensional tensor. This tensor actually represents

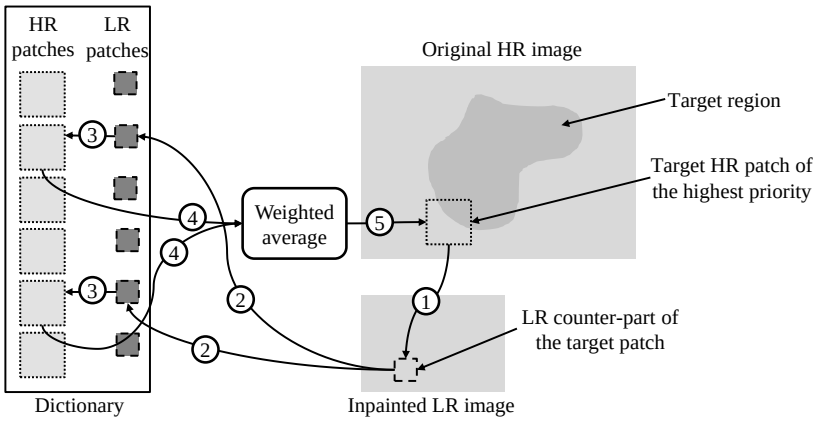


Figure 4.4: Schematic presentation of SR-based inpainting from [Le Meur 12].

a vectorized patch (assuming raster-scan order) centred at each pixel, where N determines the size of that patch. Then tensor voting is a patch matching operation, where the pixel whose tensor is the best match of the missing pixel's tensor is used to replace the missing pixel. Therefore, the method is greedy. Note that the missing region is filled pixel-by-pixel, but we still consider this method as patch-based, because eventually the surrounding patch (via ND tensor) is used for finding the candidate.

The main problem of greedy methods is that the selection of the candidate patch is based only on the known part of the target patch and hence ignoring the context, as we discussed earlier at the beginning of Section 4.3.1. This problem is further aggravated by choosing only one candidate patch (the best match) for each target patch (or pixel), which may be quite limiting. As a consequence, greedy methods are prone to errors, which then tend to propagate inside the missing region, causing visually inconsistent results. Constraining the patch search to segments, as in [Jia 03], could improve on this behaviour, because in this way the patch-selection process is guided based on some additional information, which is not limited to the small patch only. However, segmentation by itself is a difficult task and if not done properly it can lead to further deterioration of the result.

4.3.1.2 Multiple-candidate methods

Multiple-candidate methods aim at improving on some aspects of greedy methods by choosing multiple candidates for each target patch. They can be divided into *weighted-average* and *sparsity-based* methods, depending on how these candidates are combined to produce the final replacement for the missing part of the target patch.

Weighted-average-based methods typically choose several most similar source patches of each target patch as multiple candidates, and, as the

name says, replace each missing pixel in the target patch with a weighted average of corresponding candidate pixels. The method from [Wong 08] introduces this concept as a straightforward extension of the algorithm from [Criminisi 04] in the non-local (NL) means fashion [Buades 05]. This means that the weights of the K most similar source patches are computed as an exponential function of their similarity with the target patch. However, the improvement in the quality of the result compared with [Criminisi 04] is very small. Furthermore, the number of candidates is fixed and it remains the same for all target patches, which may introduce blur for textured patches. A solution to this problem, proposed in [Voronin 11, Le Meur 11, Le Meur 12], is to locally adapt this number by choosing the candidates that yield similarity within some range of the similarity of the best-matching candidate. Additionally, the method in [Voronin 11] proposes to use adaptive patch size and shape and increase the patch-search space by rotating original source patches, which additionally increases the computation time. The method from [Le Meur 11] introduces a novel priority definition (Section 4.3.3) and the directional patch search (Section 4.3.2).

The method from [Le Meur 12] combines several properties of different image inpainting methods to first inpaint the image at the low resolution (LR), which is easier and computationally less demanding. Then this result is used to recover the image at high (original) resolution (HR) by using ideas from single-image super-resolution (SR) methods [Glasner 09] (see Section 3.2.5), which we also explored in our SR method proposed in Section 3.4. In particular, a dictionary is built from pairs of LR/HR patches (taken from available LR and HR data, respectively) and the HR image is inpainted by visiting the HR target patches according to the filling order, finding multiple candidate patches of its LR counterpart within the database, and then filling the missing pixels in the HR patch with the weighted average of the HR pairs of the LR candidates (see Fig. 4.4 for graphical representation). Weights are calculated based on the similarity of both LR patches and known parts of HR patches.

Sparsity-based methods [Shen 09, Xu 10] view image inpainting as a problem of sequential incomplete signal recovery under the assumption that the target patch can be represented as a sparse linear combination of candidate patches (see [Rubinstein 10] for an overview of sparse-modelling methods). The method from [Shen 09] is also an extension of [Criminisi 04], but now the target patch is viewed as an incomplete signal, which admits a sparse combination over candidate patches. Candidate patches are either all possible source patches or some directly sampled subset, which in the end form a redundant dictionary. Then based on the known part of the target patch, sparse coefficients can be estimated and used to recover the missing part of the target patch as a sparse linear combination of dictionary elements. The method in [Xu 10] elaborates further on this approach, by proposing a novel constrained optimization algorithm that derives sparse linear combination coefficients for several most similar candidates. The introduced constraint, called local patch consistency, enforces that the recovered part of the target patch is similar to the corresponding pix-

els of its neighbouring patches. Additionally, the novel priority definition is introduced (Section 4.3.3).

Note that sparse representations were also used for inpainting in [Elad 05, Guleryuz 06a, Guleryuz 06b, Fadili 09], but we do not consider these methods as patch-based. Moreover, they were mostly used in applications where missing regions are small (e.g., long thin regions or small missing blocks), or where missing pixels are randomly distributed across the image. As noted in [Xu 10], these methods have similar behaviour like geometry-based methods, in the sense that they may fail to recover structure and texture and may introduce blur when filling in larger holes.

Multiple-candidate methods, in general, represent an improvement over the greedy methods. However, better results can be a consequence of improving other components of the algorithm, i.e., priority definition and patch search. Furthermore, for most multiple-candidate methods, the main problem of selecting patches based on the known part of a small target patch remains. The steps towards its solution are taken in [Le Meur 12], by using the inpainted LR image to guide the inpainting of the HR image, and in [Xu 10], by enforcing local patch consistency. However, in [Le Meur 12], inpainting of the LR image still suffers from the above mentioned problem, and the errors that could arise would propagate to the higher resolution level. On the other hand, in [Xu 10], local patch consistency is considered only during the sparse reconstruction step, after multiple candidates have already been chosen based *only* on the known pixels of the target patch. Between weighted-average and sparsity-based approaches, the latter group may be advantageous because it is less prone to the introduction of blur.

4.3.1.3 Global methods

Global methods treat inpainting as a global optimization problem. There are two advantages to this approach: 1) multiple candidates (called *labels*) are chosen for each location in the target region, and 2) they are selected based on both the known part of the target patch and the neighbouring information and combined within a global optimization function. In this way, a wider context is considered when inferring the missing information in the image.

Such a global approach can be applied to recover only the image structures (curves), which should be preserved by the inpainting process. The idea of this approach is similar to the one of segmentation-based inpainting [Jia 03]: to continue the curves inside the missing region, while the rest of the missing region is divided by the curves into sub-regions (or segments), which are individually filled patch-by-patch by using only candidate patches from the subregion of the current target patch (see also Section 4.3.1.1). However, instead of extrapolating partitioning curves obtained by segmentation using tensor voting as in [Jia 03], a graph can be constructed along the *user-specified* curves, as proposed in [Sun 05] (see Fig. 4.5 for graphical representation). The graph consists of nodes, which are positioned on the curves. Each node is to be assigned one of the labels, which are the patches positioned in the narrow band

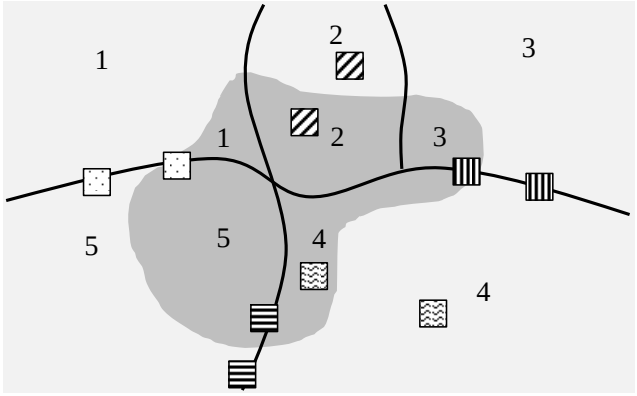


Figure 4.5: Schematic presentation of the method from [Sun 05] (see text for details). The dark grey area is the missing region and the black curves are provided by a user. The target patch (or unknown patch centred at the node if it lies on the curve) and its possible corresponding source patch (or label) are of the same pattern. Note that if the target patch belongs to some sub-region, its source patch will be from the same sub-region (e.g., patches with diagonal lines in sub-region 2 and patches with waves in sub-region 4). Also note that labels of the node, which is positioned on the curve, lie along that same curve (e.g., patches with dots, vertical and horizontal lines).

along the corresponding specified curve in the source region (e.g., patches with dots, vertical and horizontal lines in Fig. 4.5). Patches assigned to neighbouring nodes overlap. The energy function is then defined in order to enforce the agreement of the known part at the node with its labels (called the *label cost* term) and the agreement between labels of neighbouring nodes in the region of overlap (called the *consistency* or *coherence* term), where agreement is measured in terms of SSD. Additionally, structure similarity between a node and its labels is enforced in order to ensure better continuation of curves. The energy is minimized using the belief propagation (BP) algorithm [Pearl 88] (see also Section 2.3.5 for loopy version of the algorithm) in order to obtain the labelling of the graph. This method does a good job in preserving structures and it eliminates the need for segmentation, but it relies heavily on user interaction.

Global optimization can also be used for inpainting of the whole missing region by modelling global image context with an MRF (see Section 2.2). In that case, an MRF is defined over a graph consisting of nodes, which represent central positions of overlapping square masks that intersect the target region. Each node has a set of labels, which are *all* possible patches completely inside the source region. Methods that apply this approach, the so-called MRF-based inpainting methods in [Komodakis 07, Huang 07, Yang 09], mainly differ in the form of the global energy function of the MRF. In [Komodakis 07], this function consists of the label cost and consistency terms, as in [Sun 05], but no structure similarity is considered. This objective function is then optimized with a smart algorithm based on BP, called *priority* BP (p-BP), whose main

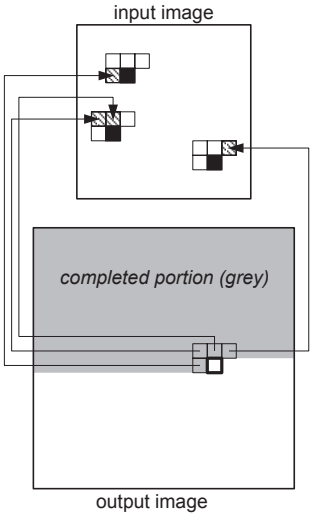


Figure 4.6: Imposing coherence in pixel-by-pixel texture synthesis algorithm (figure taken from [Ashikhmin 01]). Each pixel in the current causal neighbourhood is used for finding candidates for the current missing pixel. This is achieved by memorizing the positions of their source pixels in the input image (striped squares) and shifting them within the causal neighbourhood (black squares) so to correspond to the relative position of the current missing pixel.

purpose is to limit the number of labels of each node in a meaningful way.[3] The method from [Huang 07] combines the confidence term from [Criminisi 04] (see also Section 4.3.1.1) with the label cost and consistency terms. Moreover, structure propagation is achieved by adding a component to the consistency term that enforces the *gradients* of labels of neighbouring nodes to agree in the region of overlap. The method employs coarse-to-fine BP for optimization instead of p-BP, which is another way of reducing the number of labels. Finally, the method from [Yang 09] proposes an extension of p-BP to include structural information.

A quite different global approach, which does not employ MRF modelling, is to define some global objective function over all *missing pixels*, whose optimization produces an inpainted image. For example, in [Bugeau 10], this is achieved by minimizing an energy function over the correspondence map, which identifies for each missing pixel in the target region the location of the pixel in the source region from which the pixel value is copied. The energy is defined as a combination of three terms: 1) self-similarity, which enforces the similarity of patches centred at the target and source pixel, 2) diffusion and propagation, which are in charge of the continuation of edges, and 3) coherence, which favours that nearby corresponding points are assigned to nearby target pixels. This third term is motivated by the texture synthesis algorithm

[3]This method will be visited in more detail in Section 5.1.

from [Ashikhmin 01] (see graphical representation in Fig. 4.6). Since all three energy terms are defined as a function of the patch centred at the pixel, the method is patch-based. An interesting property of the method is the matching criterion, defined as the combination of SSD and histogram similarity between the patches. It is based on the observation that SSD alone is unable to distinguish well between smooth and textured patches. On the other hand, in [Wexler 07], a global coherence measure is proposed, which enforces that all matching patches of a patch containing a missing pixel agree on its value. Both of the aforementioned methods practically perform local optimization of the objective function for each pixel separately within an iterative process, applied at each level of the multi-resolution pyramid to speed up the convergence. The method in [Wexler 07] was developed for space-time video completion and it was later generalized for image re-targeting/re-shuffling [Simakov 08]. Although the original method is very slow and very sensitive to initialization and the optimization strategy, it is employed together with the fast patch-search method called PatchMatch [Barnes 09] (see later Section 4.3.2) within Adobe Photoshop's CS5 *Content Aware Fill*, which works well from the point of view of both quality and speed.

Finally, a very recent, advanced method from [He 12], treats image inpainting as a photomontage problem [Agarwala 04], meaning that the missing region is filled by combining a stack of shifted images. This is achieved by defining an MRF, which is optimized via multi-label graph-cut technique [Boykov 01b] (see also Section 2.3.4). MRF nodes are *all* the pixels in the target region and their labels are represented as pre-selected offsets, i.e., relative shifts compared to the location of the node. These offsets (shifts) are estimated by analysing the statistics of patch offsets, which investigates what are the most likely correspondences of the locations of similar patches within the known image region. This statistics enables one to limit the label set and guide the inpainting process (see Section 4.3.2). This method gives very good results on a variety of natural images, regardless of the size of the missing region, and it appears to be robust to the patch size. Furthermore, the reported computation time is quite low compared to the other inpainting algorithms (even compared to the Content Aware Fill of Adobe Photoshop), e.g., under 1s for images as big as 1600×1200 pixels. However, the method may run into failure when the desired offsets do not form a dominant statistics, i.e., there is insufficient number of similar patches.

4.3.2 Patch search

Many patch-based methods introduced so far, like the methods in [Criminisi 04, Komodakis 07, Wexler 07, Wong 08, Yang 09, Shen 09, Xu 10, Bugeau 10, Voronin 11], perform an *exhaustive search* for candidate patches across the whole source region of the image. This means that matching is performed for all possible source patches from the source region, which is evidently time-consuming. Some methods, e.g., in [Wexler 07], apply approximate nearest-neighbour (ANN) search [Arya 93] for faster implementation. However, this

may produce inaccurate matching result and can only be applied if the known part of the target patch is of the same size and shape and on the same position within the patch across all target patches. Moreover, exhaustive search may increase the possibility of finding a wrong match, due to the ambiguity of the known part of the target patch (see discussion at the beginning of Section 4.3.1). For global methods, exhaustive search would mean that all possible patches from the source region are considered as labels. This additionally aggravates the problem because the optimization of the global function becomes intractable.

Rather than performing exhaustive search throughout the whole inpainting process, one can employ it only in the first stages of the algorithm. The purpose is to limit the number of possible matches in subsequent stages based on the initial matches obtained by exhaustive search. For example, global methods [Komodakis 07, Yang 09], as mentioned before in Section 4.3.1.3, use a special optimization algorithm, p-BP, where the nodes are visited in some meaningful order and their unnecessary labels are discarded based on the available information at the node. This label pruning enables the definition of a computationally tractable MRF. However, prior to label pruning, all possible labels for each node are considered, making the algorithms still computationally demanding. On the other hand, the method in [Wexler 07] employs ANN search [Arya 93] only at the lowest resolution level, and locations of the matches are propagated to higher resolution levels. Finally, the method from [Bugeau 10] performs exhaustive search only in the first iteration of the algorithm, and the found K most similar patches for a patch surrounding each pixel are used as candidates in subsequent iterations.

Another solution for decreasing the computation time is to only *search in a local window* surrounding the target patch [Anupam 10,Le Meur 12], based on the observation that similar patches can be found in the immediate vicinity of the target patch. However, note that the source region remains fixed throughout the inpainting process. This means that as inpainting proceeds from the border to the center of the missing region, the number of source patches in the local window becomes smaller. Therefore, the local window becomes less reliable as a source for similar patches. Furthermore, it is difficult to determine the optimal size of this window (most likely it should be variable). Another problem is that local neighbourhood search is prone to the so-called "garbage growing" [Efros 99], in the sense that the search gets stuck at one place in the source region producing copies of one source patch for multiple locations in the missing region. The solution could be to exploit coherence, that naturally exists in images, as originally proposed in [Ashikhmin 01] (see Fig. 4.6).

In fact, this coherence has been applied in several inpainting methods, e.g., in [Bugeau 10, Le Meur 12], as well as in the fast approximate nearest-neighbour search method called PatchMatch [Barnes 09]. The idea of PatchMatch is to start from some initialization of the matches (specifically their offsets) for each patch, which can be chosen randomly or based on prior infor-

mation. Initialization is followed by an iterative procedure consisting of two steps: propagation and random search. Propagation uses coherence to improve the match by looking at matches of adjacent patches, while random search improves the solution by testing candidates taken randomly from concentric neighbourhoods of the current estimate. The method was demonstrated in applications such as image re-targeting, inpainting and re-shuffling, and in fact, it is used in PhotoShop's inpainting tool (as mentioned before).

Directional search can also be used to constrain the search for patches in a meaningful way. The idea is to perform the patch search for patches positioned on the prominent image structures in areas along those structures in the known region, as proposed in [Jia 03,Sun 05,Fang 09,Le Meur 11]. In this way, better structure propagation is enforced. The location and direction of these structures can be obtained from the user, who specifies the curves [Sun 05] (see also Section 4.3.1.3 and Fig. 4.5), or from extrapolated partitioning curves of the result of automatic image segmentation [Jia 03]. Since in this way the structures are fully specified also within the missing region, they can be used to determine the sub-regions to which the patch search is additionally constrained, depending to which sub-region a target patch belongs. Another approach of determining the position of image structures is to examine the presence and strength of an edge within the target patch based on some image features [Fang 09, Le Meur 11]. Then, if there is a strong edge, the search is performed along that e.g., [Fang 09] or the candidates in the edge direction are favoured [Le Meur 11].

Another way of avoiding exhaustive search, and thus reducing considerably computation time, is to employ *cluster-based search* [Fang 09,Huang 07]. This approach requires that all the possible candidate patches are grouped into several clusters. In order to find the best-matching source patch of a target patch, first the best-matching cluster center is identified, and then the search is performed exhaustively among the candidate patches within the corresponding cluster. This approach is applied, e.g., in [Fang 09] (see also Section 4.3.1.1), where a trained database, organized in clusters of weight vectors, is searched for the best-matching weight vector of the weight vector corresponding to the target patch. Then the source patch corresponding to that best-matching vector is used to fill in the missing pixels. MRF-based global method from [Huang 07] uses a similar cluster-based approach within a two-step inference method. In the first step, the LBP inference algorithm [Yedidia 01b] is applied using cluster centres as labels, where these centres are obtained by K-means clustering [Duda 73] of all the possible labels. This results in some labelling of the MRF. The second step comprises of running LBP again, but now the labels of each MRF node are coming from the chosen cluster.

Finally, a very recent global method from He and Sun [He 12] attempts to limit the label set by analysing the *statistics of patch offsets*. In particular, they find the best-matching patch for all the patches from the source region and they make a histogram of the occurrences of all the offsets, i.e., relative positions. What can be concluded from this statistics is that offsets

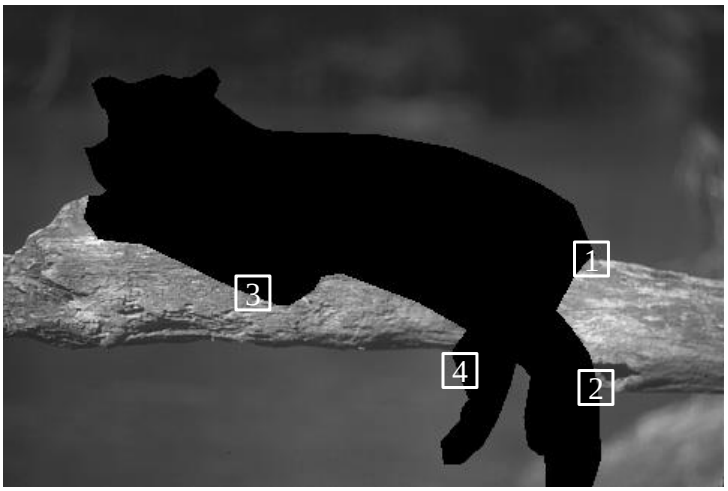


Figure 4.7: Illustration of the filling order. The numbers inside the patches indicate their filling order (from first to last). The black area marks the missing region.

are sparsely distributed and that several highest peaks in the histogram correspond to several dominant orientations. These dominant orientations not only provide reliable information for completing the image, but can also improve the quality of the inpainting result.

4.3.3 Priority definition

When inpainting an image, the goal is to fill in the missing region from its borders to the center, by first continuing image structures, i.e., lines, contours, edges, etc. This filling order is determined by priority, whose appropriate definition is necessary in order to preserve image structures. Therefore, priority should be designed to give advantage to the target patches containing structure and additionally fewer missing pixels. The second requirement ensures that the known part of the target patch is sufficiently big in order to find appropriate matches. This is illustrated in Fig. 4.7, where the black area marks the missing region. The patches marked with 1 and 2 have the highest priority because they contain structure, but the patch marked with 1 has additional advantage because it contains fewer missing pixels. Between the patches marked with 3 and 4, the patch marked with 3 has higher priority because it contains texture, while the the patch marked with 4 is flat. In the case of global methods using p-BP [Komodakis 07, Yang 09], filling order is actually the visiting order of MRF nodes according to which the labels of the nodes are pruned. However, some methods, e.g., [Huang 07, Wexler 07, Bugeau 10, He 12], do not explicitly define priority, but impose structure propagation in a different way, e.g., by defining energy terms that enforce this behaviour.

Most of the patch-based methods described in the previous subsec-

tions define priority as a product of the *confidence term* and the *data term* for each target patch at the border of the missing region, as initially proposed in [Criminisi 04] (see Section 4.3.1.1). The confidence term measures the relative amount of known pixels in the target patch, while the data term is actually responsible for encouraging the linear structures to be synthesized first, i.e., for preserving structure continuation. The data term is determined based on the strength of isophotes, i.e., gradients, and their relative direction compared to the the border of the missing region (e.g., if the isophote is orthogonal to the border, the target patch containing it has the bigger data term and thus higher priority). This priority is also adopted in other inpainting methods, e.g., in [Wong 08, Shen 09, Voronin 11]. In [Cheng 05, Anupam 10], it was noted that the confidence term decreases exponentially as inpainting proceeds towards the center of the hole. To mitigate this problem, it was suggested to replace the product of the confidence and data term by their weighted sum. However, the same gradient-based approach for data term is used, which performs insufficiently well in discriminating between patches containing structure and texture.

One solution to this problem is to find a better estimate of *local geometry*, which can better differentiate between different types of patches. Examples include Hessian matrix decision value (HDMV) [Fang 09] and structure tensor [Di Zenzo 86, Le Meur 11]. HDMV represents the ratio between eigenvalues of a Hessian matrix constructed in a small window surrounding each pixel at the border of the missing region. According to the eigenvalues and the HDMV, three types of window content can be distinguished: 1) a structured window (when HDMV is high, i.e., one eigenvalue is significantly higher than the other), 2) a textured patch (HDMV is close to or equal to one and the two eigenvalues are similar and high), and 3) a smooth patch (HDMV is close to or equal to one and both eigenvalues are low). Additionally, HDMV is used to make the choice between directional and non-directional search (see Section 4.3.2). Similar discrimination can be achieved by comparing eigenvalues of a structure tensor. Structure tensors can also be smoothed by the Gaussian filtering in order to improve robustness to noise and local orientation singularities. The latter is additionally improved in [Le Meur 11] by using a hierarchical approach, where structure tensor is propagated from lower to higher resolution levels.

Instead of estimating local geometry, one could use similarity between the target patch and the candidate patches to define priority. The reasoning is the following: if there are few well-matching patches of the target patch, it is more likely that the target patch contains structure, thus it should be assigned higher priority, and vice versa, if there are many well-matching patches, the target patch is probably textured or smooth, and it should have lower priority. Implicitly, this also means that the target patch of higher priority has more known pixels, because such patch will also have less well-matching patches. In [Le Meur 12], it was demonstrated that this approach can better distinguish structure and texture than the gradient-based priority discussed above, and

that it is more robust to the orientation of the border of the missing region. This *similarity-based priority* is practically defined in [Xu 10] via the so-called *structure sparsity*, which measures the sparseness of the similarities of the target patch within its neighbourhood (bigger sparseness means less similar patches). Additionally, this term is combined with the confidence term from [Criminisi 04] as the final patch priority. On the other hand, in [Komodakis 07, Yang 09], priority is a function of beliefs, where belief is one of the terms in the LBP algorithm (see Section 2.3.5). Beliefs can be used to define confidence of a node about its labels. The more confidence a node has about its labels, meaning there are fewer labels with belief higher than some threshold, the higher is its priority, and in practice, it is the node lying on an object border and having more known pixels (see Section 5.1.2 for details).

4.4 Context-aware approach for inpainting

In this section, we propose a general context-aware approach for patch-based image inpainting, which can be used with any patch-based inpainting algorithm. The main idea is to guide the search for patches to the areas of interest based on contextual features. Fig. 4.8 illustrates this concept: contextual descriptors are assigned to image blocks of fixed size. For the missing region within a given block, well-matching candidate patches will be found in the contextually similar blocks. We employ Gabor-based texture descriptors (see Appendix A) as contextual descriptors and extend them with colour information.

In this way, we employ the wider context around the target patch into the patch-selection process instead of just observing a small patch and its known pixels like most greedy methods and multiple-candidate methods do. Furthermore, we perform guided search for patches, which represents an alternative to the time-consuming exhaustive search. The benefit of our context-aware approach is, therefore, twofold: the search for candidate patches is accelerated and the inpainting result is improved.

4.4.1 Notations and definitions for patch-based inpainting

In this subsection, we extend the general notations and definitions for patch-based methods, introduced previously in Section 3.3, to the problem of patch-based inpainting.

In image inpainting, a damaged colour image g, over a set of pixel positions I, consists of a missing and a known region. Let $\Omega \subset I$ denote the missing region, i.e., the region to be filled, called the *target* region, and $\Phi \subset I$ denote the known (undamaged) part of the image, called the *source* region, where $\Omega \cap \Phi = \emptyset$ and initially $\Omega \cup \Phi = I$. Recall that pixel positions are represented by a single index $p \in I$, assuming raster-scan order.

Analogously to the definition of a mask Ψ in Section 3.3 (see also Fig. 3.9), we can define a square block B as a set of positions p centred at

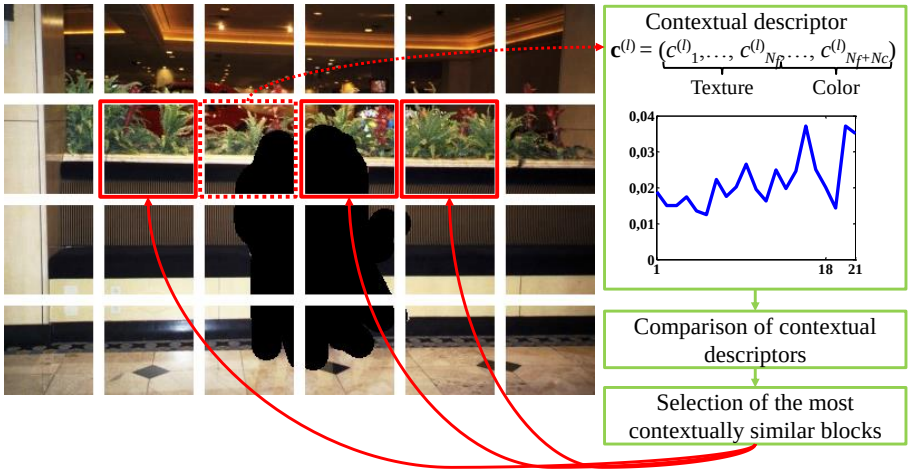


Figure 4.8: Illustration of the proposed context-aware approach for inpainting.

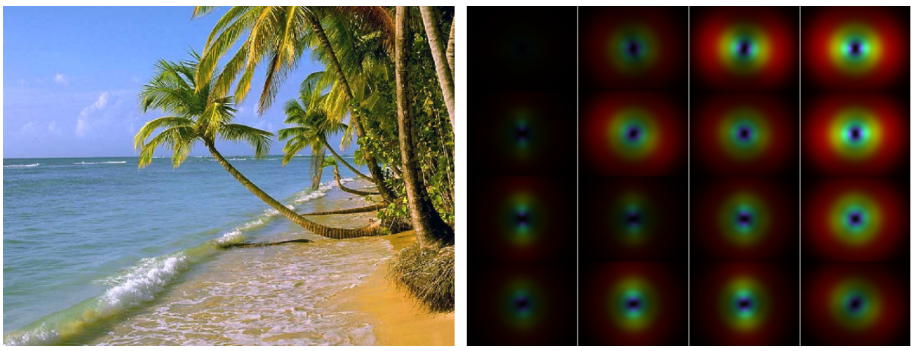


Figure 4.9: Input image (left) and its gist descriptor (right). In the right image, the average of the output magnitude within each non-overlapping block is shown. The orientation on the polar plot is orthogonal to the direction of the edges in the image, scale is colour coded (red for high spatial frequencies and blue for low), and the intensity of the colour is proportional to the magnitude of each filter output. In this example, we used Gabor filters of 8 orientations and across 4 scales.

the origin $p = 0$. A block is bigger than a mask. Let $B + l$ denote a block translated to position $l \in \Theta$, where Θ is the set of allowed block positions defined in such a way that neighbouring blocks are not overlapping. A block, in general, can contain both known and missing pixels. Let $\zeta(\cdot)$ be a function that for each pixel position returns the central position of the block to which that pixel belongs. Hence, $B + \zeta(p)$ is the block containing the pixel p.

4.4.2 Context representation

Suppose we divide the image into $N_v \times N_h$ square non-overlapping blocks of fixed size, then $n_b = N_v N_h$ is the total number of blocks (see Fig. 4.8 and the top of Fig. 4.10). Using the notations for blocks we introduced above, we can refer to each of the n_b blocks by their central positions $l \in \Theta$. The set Θ of block positions, as well as the block size, are determined by the division of the image into blocks. We want to assign a contextual descriptor $\mathbf{c}^{(l)}$ to each block $B + l$, which characterizes in some way spatial content and textures within that block. Moreover, we want to have compact descriptors that allow efficient search for regions of similar context.[4]

In this chapter, we propose a combination of texture and colour features as contextual descriptors. The choice of texture features, as well as the simple division of the image into blocks of fixed size, are motivated by a global image representation called *gist* [Oliva 01], due to its simplicity and efficiency. Gist is computed by averaging the magnitudes of filter responses within square non-overlapping image blocks, where filter responses are obtained by filtering the luminance channel of the image with a bank of multi-scale oriented filters, e.g., Gabor filters (see Section A.3 for details). Fig. 4.9 shows the information that gist descriptor provides: a coarse description of textures in the image and their spatial organization.[5]

Instead of using gist as a global image representation, for the inpainting purpose we will focus on the coarse texture description *per block*. Such description gives an idea of the context surrounding pixels or patches within that block. In Fig. 4.9, we can see that the blocks with similar content have similar representation (e.g., blocks containing parts of the palm trees), which is exactly the property we want to use for our context-aware approach. While gist was used as a scene descriptor, e.g., for scene classification [Oliva 01], we do not know of any methods in which it was used for image inpainting.

Let $\mathbf{f}(p) = (f_1(p), \ldots, f_{N_f}(p))$ denote an N_f-dimensional vector of *complex* filter responses[6] at pixel $p \in I$, obtained by filtering the image with the bank of N_f multi-scale oriented complex Gabor filters (see Section A.2 for details on Gabor filtering). We define the contextual descriptor $\mathbf{c}^{(l)}$ of the block $B + l$ as an $(N_f + N_c)$-dimensional feature vector, where the first N_f components represent texture descriptor, and the last N_c components represent colour descriptor (see also Fig. 4.8). The first N_f components of $\mathbf{c}^{(l)}$ are related to gist, and are computed by averaging the magnitudes of complex filter responses at the *known* pixels of the block $B + l$:

[4] Alternatively, contextual descriptors could be assigned to overlapping blocks, but we have not seen any benefits of such division in our experiments, in terms of contextual similarity.

[5] Gist descriptors are obtained with the source code available at http://people.csail.mit.edu/torralba/code/spatialenvelope/.

[6] Note that in Chapter 3, $\mathbf{f}$ denoted a vector formulation (assuming raster-scan order) of a high-resolution image f.

$$c_n^{(l)} = \frac{1}{\#((B+l)\cap\Phi)} \sum_{p\in((B+l)\cap\Phi)} |f_n(p)|, \quad n = 1,\ldots,N_f, \tag{4.1}$$

where # denotes the cardinality of the set. This texture descriptor contains average information about the presence of image energy at certain orientations and at certain scales within the block.

The last N_c components of $\mathbf{c}^{(l)}$, i.e., the colour descriptor, represent the average colour within the block $B+l$ per each colour channel:

$$c_{N_f+n}^{(l)} = \frac{1}{\#((B+l)\cap\Phi)} \sum_{p\in((B+l)\cap\Phi)} g_n(p), \quad n = 1,\ldots,N_c, \tag{4.2}$$

where $g_n(p)$ is the colour value at pixel position p in the n^{th} colour channel and $N_c = 3$. The averaged colour values per channel are typically higher than the averaged filter responses. Hence, we normalize the colour components by the factor α, $c_{N_f+n}^{(l)} = \alpha^{-1} c_{N_f+n}^{(l)}$, $n = 1,\ldots,N_c$. We define this factor as the ratio between the maximum value of the last N_c colour components and the maximum value of the averaged filter responses on the first N_f components:

$$\alpha = \frac{\max_{n\in\{N_f+1,\ldots,N_f+N_c\}} c_n^{(l)}}{\max_{n\in\{1,\ldots,N_f\}} c_n^{(l)} + \epsilon}, \tag{4.3}$$

where ϵ is a small constant that prevents division by zero. Throughout this chapter, we use complex Gabor filters of 6 orientations and across 3 scales (see Fig. A.1), thus $N_f = 18$. The resulting feature vector $\mathbf{c}^{(l)}$ then has 21 components and it shows dominant orientations and scales within the block $B+l$ and the average colour of that block. Fig. 4.10 illustrates these feature vectors corresponding to different blocks of an image. The first 18 components (texture features) are ordered by orientation per scale, from small to large scales, i.e., from high to low spatial frequencies (see the first 18 components in the plot in Fig. 4.8 and in the bottom of Fig. 4.10). We can see that the texture features are small for nearly flat blocks (most of the blocks in the second and third row in the bottom of Fig. 4.10). For blocks with dominant edges (in the last two rows), the peaks appear at positions corresponding to a particular orientation and tend to increase when the scale coarsens. Textured blocks (in the fourth row for example) have smaller descriptor values and smaller peaks at multiple orientations.

4.4.3 Context-aware patch selection

Now that we chose a context representation, we want to constrain the source region for target patches, belonging to some current block $B+l$, to a region $\Phi^{(l)} \subset \Phi$ with a context well matching that of $B+l$. Let $\bar{H}^{(l,m)}$ denote a

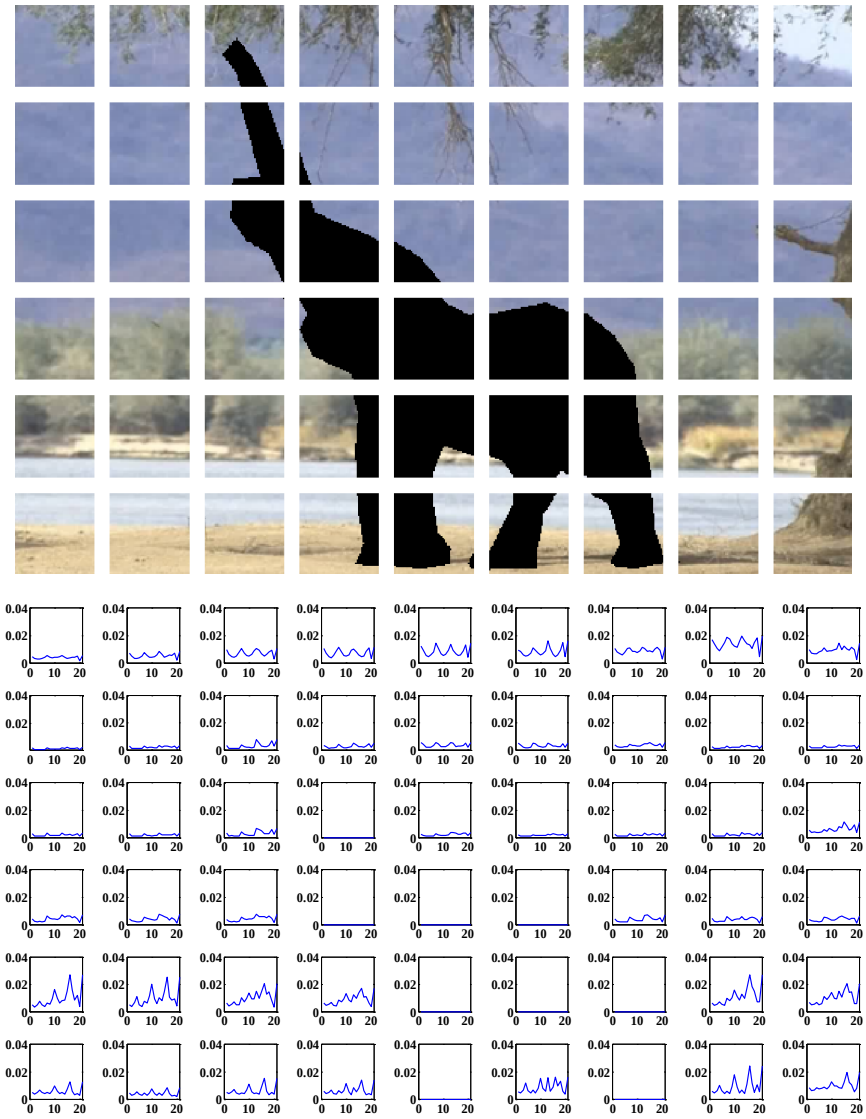


Figure 4.10: Top: division of the image into 6×9 non-overlapping blocks. Bottom: corresponding contextual descriptors plotted over 21 components and with values ranging between 0 and 0.04. The first 18 components, corresponding to texture features, are ordered by orientation per scale, from high to low scales, i.e., high to low spatial frequencies. Contextual descriptors of unreliable blocks (see later Eq. (4.7)) are set to zero.

measure of contextual *dissimilarity* between blocks $B+l$ and $B+m$. We define $\bar{H}^{(l,m)}$ as some distance measure between contextual descriptors (vectors) $\mathbf{c}^{(l)}$ of the block $B+l$ and $\mathbf{c}^{(m)}$ of the block $B+m$:

$$\bar{H}^{(l,m)} = d(\mathbf{c}^{(l)}, \mathbf{c}^{(m)}). \tag{4.4}$$

In this chapter, we take this distance measure to be the SSD between the vectors $\mathbf{c}^{(l)}$ and $\mathbf{c}^{(m)}$:

$$\bar{H}^{(l,m)} = \sum_{n=1}^{N_f+N_c} (c_n^{(l)} - c_n^{(m)})^2. \tag{4.5}$$

The choice of this measure was motivated by the comparison of gists of two images for scene completion in [Hays 08].

Let $\Sigma^{(l)}$ denote the set of positions of blocks $B+m$ that are contextually similar to the current block $B+l$. We define contextually similar blocks as the ones that yield Z smallest contextual dissimilarities $\bar{H}^{(l,m)}$. The *constrained source region* $\Phi^{(l)}$ is then a union of *known* parts of blocks $B+m$, which are contextually similar to the block $B+l$, i.e., whose contextual descriptors are close to that of $B+l$:

$$\Phi^{(l)} = \cup_{m\in\Sigma^{(l)}}((B+m)\cap\Phi). \tag{4.6}$$

Note that the current block itself is always a part of $\Phi^{(l)}$ because $\bar{H}^{(l,l)} = 0$. In practice, the number of chosen blocks Z is proportional to the number of blocks in the image, $Z = n_b/r$, where the fraction r is some constant.

In practical applications, however, some blocks are dominated by missing pixels (e.g., central blocks in Figs. 4.8 and 4.10) and hence yield unreliable contextual descriptors. To indicate whether a block contains enough information from which the contextual descriptors can be drawn, we define the block *reliability* $\rho^{(l)}$ as

$$\rho^{(l)} = \begin{cases} 1, & \text{if} \quad \#((B+l)\cap\Phi) > \frac{\#(B+l)}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{4.7}$$

If $\rho^{(l)} = 1$, the block $B+l$ is reliable and vice versa, it is unreliable if $\rho^{(l)} = 0$. The contextual descriptor of an unreliable block is also unreliable and cannot be used directly to find contextually similar blocks. Instead, we use its neighbouring blocks $B+l'$ to define the constrained source region $\Phi^{(l)}$. Therefore, to account for both reliable and unreliable blocks, we express Eq. (4.6) in a more general form as

$$\Phi^{(l)} = \begin{cases} \cup_{m\in\Sigma^{(l)}}((B+m)\cap\Phi), & \text{if} \quad \rho^{(l)} = 1 \\ ((B+l)\cap\Phi) \cup (\cup_{\substack{l'\in\partial l \\ \rho^{(l')}=1}} \Phi^{(l')}), & \text{if} \quad \rho^{(l)} = 0, \end{cases} \tag{4.8}$$

where ∂l denotes the neighbourhood of l, i.e., the set of central positions of the blocks neighbouring the block $B+l$. According to Eq. (4.8), the constrained source region $\Phi^{(l)}$ of the current *unreliable* block $B+l$ consists of the known part of the block $B+l$ itself and block matches of all of its neighbouring reliable

Algorithm 1 Context-aware patch selection

1: **for all** $B+l$ such that $l \in \Theta$ and $(B+l) \cap \Omega \neq \emptyset$ **do**
2: compute reliability $\rho^{(l)}$ of the block $B+l$ (Eq. (4.7))
3: **if** $\rho^{(l)} = 1$ **then**
4: compute $\bar{H}^{(l,m)}, \quad \forall m \in \Theta$
5: define $\Sigma^{(l)}$ as the set of positions of contextually similar blocks to the block $B+l$
6: define new source region $\Phi^{(l)}$
7: **else**
8: **for all** $B+l'$ such that $l' \in \partial l$ **do**
9: repeat steps 2-6
10: **end for**
11: define new source region $\Phi^{(l)}$
12: **end if**
13: **end for**

blocks ($\cup_{\substack{l' \in \partial l \\ \rho^{(l')}=1}} \Phi^{(l')}$). The proposed context-aware approach is summarized as pseudo-code in Algorithm 1.

Fig. 4.11 illustrates block-matching results for current blocks from Fig. 4.8 (shown in Fig. 4.11(a)), by using *only* texture descriptors, i.e., the first N_f components of contextual descriptors (Fig. 4.11(b)), and both texture and colour descriptors, i.e., the complete contextual descriptors (Fig. 4.11(c)). For example, in the last row, we can see that both choices of descriptors can find contextually similar blocks. However, the first three rows clearly demonstrate that using colour improves the block-matching performance. Some of the mismatches (previously inside the blue square) are now eliminated and replaced with better matches.

A few implementation details are described next. The neighbourhood ∂l initially consists of the top, bottom, left and right neighbours of the current block $B+l$. If none of these neighbours is reliable (which could happen if the block size is too small relative to the size of the target region), the neighbourhood is extended to the diagonal neighbours. Furthermore, note that a target patch can span multiple blocks, e.g., $B+l$ and $B+l'$, in which case the constrained source region for that patch would be $\Phi^{(l,l')} = \Phi^{(l)} \cup \Phi^{(l')}$. Finally, although our contextual descriptors are computed within non-overlapping blocks, we take source patches from a block extended by w in each dimension, where $2w+1 \times 2w+1$ is the patch size, by requiring the central pixel of the patch to belong to the constrained source region (see later Eq. (4.15)). In this way, also the source patches spanning multiple blocks are considered.

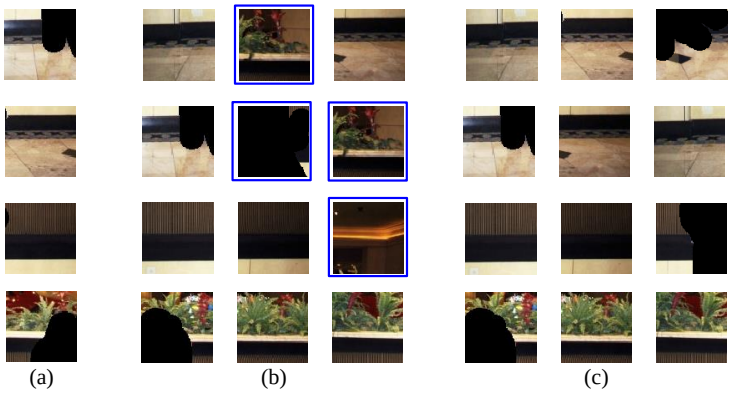


Figure 4.11: Block matches: (a) current blocks from Fig. 4.8, (b) block matches based on texture descriptors, and (c) block matches based on both texture and colour descriptors. Each row represents block matches of the block in (a). Blocks are ordered by the increasing block-matching error. The blue squares mark the mismatches.

4.5 Proposed context-aware inpainting method

In this section, we apply the proposed context-aware approach within a novel context-aware inpainting algorithm. In addition to context-awareness, the proposed algorithm explores the use of contour features [Perona 90, Malik 01] to define the patch priority. We propose a novel priority definition, called *orientation-based* priority, where the goal is to achieve better differentiation between patches containing contours, textured patches and patches in flat areas compared to the gradient-based patch priority of [Criminisi 04] (see also Section 4.3.3). Therefore, our proposed method aims at improving all three components of patch-based inpainting algorithms: patch selection and patch search, through context-awareness, and priority definition.

4.5.1 Orientation-based priority

Let $\delta\Omega$ denote the border of the target region. The filling order is defined by the priority $R(p)$ for each of the target patches centred at $p \in \delta\Omega$ (see Section 3.3 for the definition of the patch).

As we discussed earlier in Section 4.3 and in more detail in Section 4.3.3, the filling order of the missing region, defined via priority, is an important component of patch-based inpainting algorithms. Priority is defined in a way that ensures the propagation of image structures inside the missing region, thus often involving the detection of these structures at the central pixel of each target patch by the means of gradient computation. However, we have at our disposal filter outputs due to our context-aware approach, which explores texture features obtained by filtering the image with the bank of Gabor filters at multiple orientations and scales. Analysis of filter outputs can

Figure 4.12: Binary images indicating the central lines of a 15×15 mask with orientations from 0° to 150° in steps of 30°.

provide a better estimate of local geometry than gradients because of the ability to detect more complex edges by extracting *contour features* (see also the discussion in Section A.4). A well-known approach for the extraction of contour features, namely dominant orientation and oriented energy, is the *oriented energy approach* [Perona 90] (see Section A.4 for more detail).

The oriented energy $OE_\theta(p)$ at the pixel $p \in I$ represents the strength of filter responses in each orientation θ, which is orthogonal to the filter orientation η. Let $f_{\theta,\varsigma}(p)$ denote the complex filter response to the complex Gabor filter $G_{\eta,\varsigma}(p)$. The oriented energy is then defined as

$$\begin{aligned} OE_\theta(p) &= [\mathrm{Re}\big(f_{\theta,\varsigma}(p)\big)]^2 + [\mathrm{Im}\big(f_{\theta,\varsigma}(p)\big)]^2 \\ &= |f_{\theta,\varsigma}(p)|^2, \end{aligned} \tag{4.9}$$

Then the dominant orientation $\theta^*(p)$ at p is defined as

$$\theta^*(p) = \arg\max_\theta OE_\theta(p), \tag{4.10}$$

and it represents the orientation of the strongest contour at p, and the strength of that contour is $OE_{\theta^*}(p)$. $OE_{\theta^*}(p)$ additionally undergoes non-maximal suppression [Canny 86] for better localization of the contour, resulting in the non-maximal suppressed oriented energy $OE^*(p)$ (see Eq. (A.5) and Section A.4 for more detail).

We use these contour features to define a novel priority, called orientation-based priority, as a combination of *contour strength* and *directional strength* of each target patch. The contour strength $D_{con}(p)$ of the patch centred at p is represented by the non-maximal suppressed oriented energy at the central pixel p:

$$D_{con}(p) = OE^*(p). \tag{4.11}$$

We determine the directional strength $D_{dir}(p)$ via the statistics of dominant orientations at the known pixels of the patch. Let us define $L_\gamma + p$ as a set of positions on the central line of the mask $\Psi + p$, where γ denotes the orientation of the line. These positions are indicated in white in the binary images shown in Fig. 4.12. The orientation of the line $\gamma \in \Gamma$ coincides with contour orientation, thus $\Gamma = \{0°, 30°, 60°, 90°, 120°, 150°\}$ (because we use complex Gabor filters of 6 orientations). To evaluate the statistics of dominant orientations within the patch centred at p, it is sufficient to consider only the pixels in $L_\gamma + p$, i.e., along the thin lines shown in Fig. 4.12, because we perform this evaluation for

each patch along the border of the missing region, i.e., centred at each pixel $p \in \delta\Omega$. Therefore, the pixels on the lines parallel to the lines shown in Fig. 4.12 will be considered while evaluating the statistics of the neighbouring patches.

We define the directional strength of the patch centred at p as the maximum over $\gamma \in \Gamma$ of the relative number of pixels $p' \in ((L_\gamma + p) \cap \Phi)$, at which the dominant orientation $\theta^*(p')$ coincides with the orientation γ, i.e.,

$$D_{dir}(p) = \max_{\gamma \in \Gamma} \frac{\#\{p' \in ((L_\gamma + p) \cap \Phi) | \theta^*(p') = \gamma\}}{\#((L_\gamma + p) \cap \Phi)}, \tag{4.12}$$

where $\theta^*(p')$ is defined in Eq. (4.10). The reasoning behind this choice is the following. A high value of the directional strength indicates that the pixels along certain direction within the target patch lie on the same contour (see the last row of Figs. 4.13(a) and (b)), thus it is more probable that the target patch contains image structure. On the other hand, if this value is low (Figs. 4.13(c) and (d)), it means that the dominant orientations at pixels along all directions are different, i.e., it is more probable that the target patch belongs to the textured region. Finally, very low or zero value of the directional strength (Fig. 4.13(e)) indicate that the target patch is flat.

When we evaluate $D_{dir}(p)$, we do not consider pixels for which $OE^*(p) < T_{OE}$ (pixels in dark red in the bottom row of Fig. 4.13). If the target patch centred at p consists only of such pixels, e.g., in flat areas, we set its directional strength to zero, i.e., $D_{dir}(p) = 0$. This is because non-maximal suppression (Eq. (A.5)) sets the oriented energy and dominant orientation of certain pixels to zero. This orientation coincides with the horizontal one, while, in fact, it should not be specified. Furthermore, we want to discard from analysis all the pixels with very low values of orientation energy to obtain more stable measurements.

The directional strength is necessary because the contour strength alone is not robust enough to distinguish between different types of patches. For example, $D_{con}(p)$ (see the middle row of Fig. 4.13) can have higher value for the textured patch (Fig. 4.13(c)) than for the patch containing contour (Fig. 4.13(a) and (b)). On the other hand, it can also be zero for the textured patch due to the non-maximal suppression (Fig. 4.13(d)), which should normally happen in the flat area (Fig. 4.13(e)). To circumvent this problem, we could, as in [Criminisi 04], look at the relative orientation between the contour and the border of the target region, but this approach is not robust to the orientation of the border and relies significantly on the measurements at one (central) pixel of the patch. On the other hand, the proposed directional strength does not suffer from these problems.

In the end, $D_{con}(p)$ and $D_{dir}(p)$ are combined in one *orientation-based priority* as

$$R(p) = \frac{D_{con}(p) + \mu D_{dir}(p)}{2}, \tag{4.13}$$

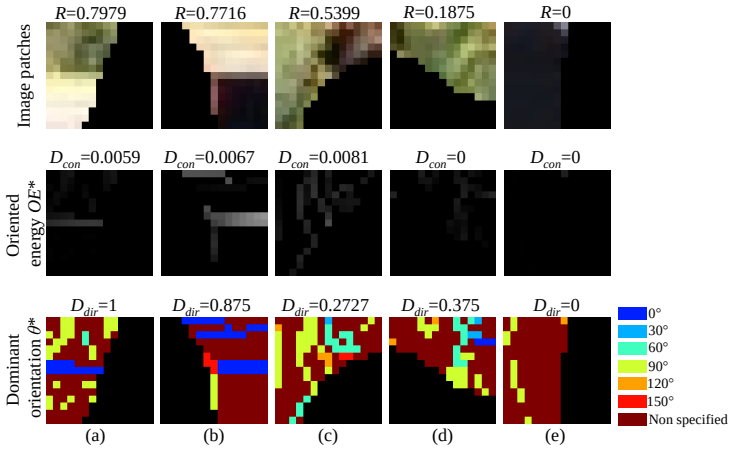


Figure 4.13: 15×15 patches from Fig. 4.8 with the missing region in black: colour image patches (top row), corresponding oriented energies OE^* (middle row), and corresponding dominant orientations θ^* (bottom row). In the bottom row, "non-specified" dominant orientation marks the pixels that are not considered in the analysis (see text for details). Corresponding values of R, D_{con} and D_{dir} are indicated.

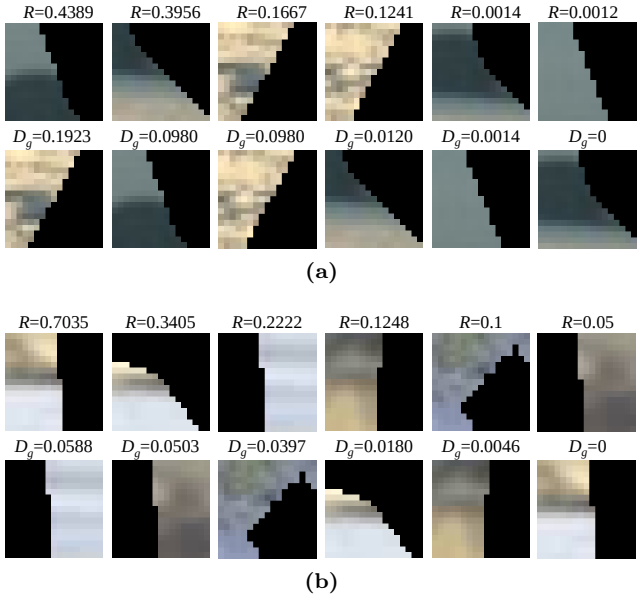


Figure 4.14: Comparison between the proposed priority R and the gradient-based data term D_g from [Criminisi 04]: (a) patches from Fig. 4.16, and (b) patches from Fig. 4.17. Patches are ordered from left to right by decreasing value of R (top row) and D_g (bottom row).

where μ is the weighting factor that ensures that the two components are in the same range of values. Such a priority distinguishes better between patches with contour and textures than the gradient-based definition of the data term, as demonstrated in Fig. 4.14 for patches from two different images. We can see that the proposed priority R has higher value for the patch containing contour than for the textured patch, while the gradient-based data term, denoted as D_g, would give preference to the textured patch.

4.5.2 Greedy block-based context-aware (GBCA) inpainting method

In this subsection, we present the actual inpainting method that incorporates two novelties introduced earlier in this chapter: context-aware patch selection (Section 4.4.3) and orientation-based priority (Section 4.5.1). We fill in the missing region iteratively, where in each iteration for the target patch of the highest orientation-based priority, we find the *best* match from the *constrained* source region, based on context-aware patch selection, and we copy its corresponding pixels to the locations of missing pixels. We name this method greedy block-based context-aware (GBCA) method because it selects only the best match in a greedy manner.

Let us assume that at this point we have already extracted contour features for each pixel, i.e., oriented energy and dominant orientation, and determined contextual descriptors for each block. The inpainting algorithm proceeds in the following manner until there are no more missing pixels. At each step t, the border $\delta\Omega^{(t)}$ is identified and the priorities $R(p)$ are computed as in Eq. (4.13), $\forall p \in \delta\Omega^{(t)}$. Then we find the target patch with the highest priority as the one whose central pixel is

$$\hat{p} = \arg \max_{p \in \delta\Omega^{(t)}} R(p). \tag{4.14}$$

In order to perform context-aware patch selection, introduced earlier in Section 4.4.3, we need to find a block to which the pixel $\hat{p}$ belongs by using the function $\zeta(\cdot)$ introduced in Section 4.4.1. Hence, $B + \zeta(\hat{p})$ is the block containing the pixel $\hat{p}$. Context-aware patch selection then constrains the source region for the target patch centred at $\hat{p}$ to $\Phi^{(\zeta(\hat{p}))} \subset \Phi$ (see Algorithm 1 for details). The best-matching patch of the target patch is found in this constrained source region by calculating the SSD only between the *known* pixels of the target patch and the corresponding pixels of the candidate patch. Using the notations introduced in Section 3.3, the central pixel of the best-matching patch is defined as:

$$\hat{q} = \arg \min_{q \in \Phi^{(\zeta(\hat{p}))}} \|\mathcal{T}_{\hat{p}} g - \mathcal{T}_q g\|^2_{((I \setminus \Omega^{(t)}) - \hat{p}) \cap \Psi}. \tag{4.15}$$

The shape $((I \setminus \Omega^{(t)}) - \hat{p}) \cap \Psi$ is obtained by translating the current set of positions of known pixels $I \setminus \Omega^{(t)}$ so that $\hat{p}$ is now at the origin, and then finding the intersection with the mask Ψ centred at the origin. Thus this

Algorithm 2 GBCA inpainting method

while $\Omega^{(t)} \neq \emptyset$ **do**
    identify the fill front $\delta\Omega^{(t)}$
    compute priorities $R(p), \quad \forall p \in \delta\Omega^{(t)}$ (Eq. (4.13))
    find the central position $\hat{p}$ of the patch with the highest priority (Eq. (4.14))
    find the current block $B + \zeta(\hat{p})$ to which $\hat{p}$ belongs
    find the constrained source region $\Phi^{(\zeta(\hat{p}))}$ with Algorithm 1
    find the the best-matching patch of the target patch (Eq. (4.15))
    replace missing pixels of the target patch
    update $\Omega^{(t)}$, $\mathbf{f}$, $\mathbf{c}^{(\zeta(\hat{p}))}$, OE^* and θ^*
end while

shape corresponds to the current mask of the known pixels of the target patch centred at $\hat{p}$. Finally, the missing pixels in the target patch are replaced with the corresponding ones from the best match centred at $\hat{q}$.

Replacing missing pixels of the target patch means that the target region is shrinking at each step t, i.e., that $\Omega^{(t)}$ needs to be updated, while the source region Φ remains the same throughout the whole algorithm, thus $I \setminus \Omega^{(t)} \neq \Phi$. However, we still determine the constrained source region $\Phi^{(\zeta(\hat{p}))}$ at each step, because as the filling process proceeds, there is more available contextual information. We obtain this additional information by updating the filter responses $\mathbf{f}$, i.e., by copying the corresponding magnitudes of filter responses from the found source patch. Then we can update the contextual descriptor $\mathbf{c}^{(\zeta(\hat{p}))}$ of the current block $B + \zeta(\hat{p})$. Such update yields more reliable contextual descriptors and better block-matching result. Furthermore, once an unreliable block becomes reliable, we can use its own contextual descriptor for matching instead of contextual descriptors of neighbouring blocks (see Eqs. (4.7) and (4.8)), leading to the context being better determined and saving some computation time because the constrained source region is smaller. The pseudo-code of the algorithm is shown in Algorithm 2. Note that also OE^* and θ^* need to be updated by again copying the corresponding values from the found match, in order to have information for priority computation.

4.6 Results

We tested the proposed context-aware inpainting method on a number of natural images and one artificial example. We consider the application of image editing, which involves object removal and is typically more demanding than, e.g., text removal, due to the size of the missing region. We compare the proposed method with different patch-based inpainting methods. For the proposed and all the reference methods we show the best inpainting result, depending on the patch size. Our method also depends on the division into blocks, so we also chose the division that yields the best inpainting result. The number of

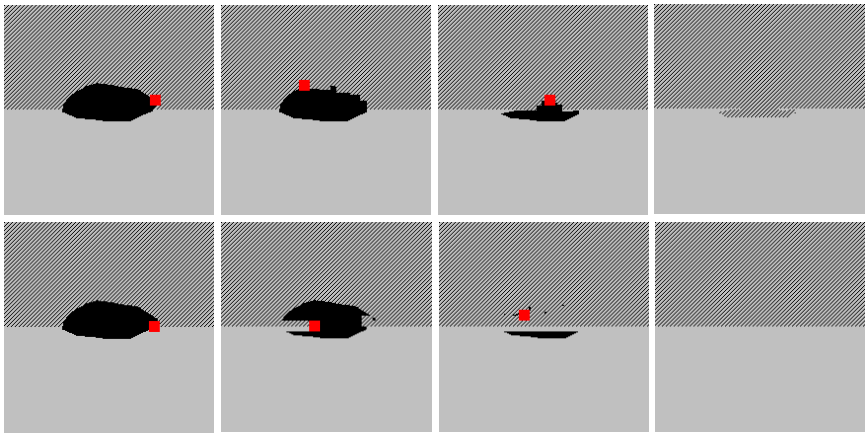


Figure 4.15: Dependence of inpainting result on priority definition. Top: inpainting process with the original priority as in [Criminisi 04]. Bottom: inpainting process with the proposed priority definition. From left to right: results after iterations 1, 10, 50, and final result. Current target patch is marked with red.

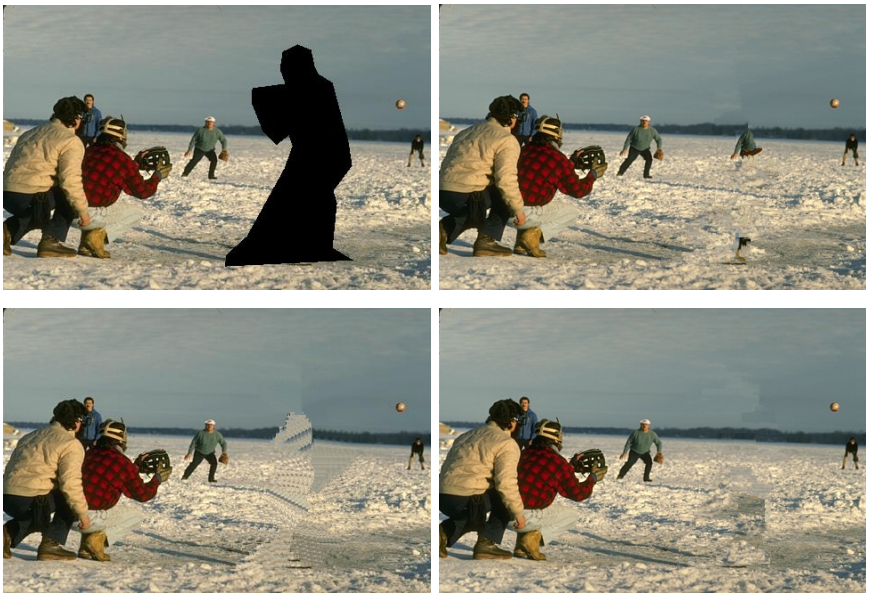


Figure 4.16: Comparison of inpainting results for the "baseball" image. From left to right and top to bottom: input image with the missing region marked in black, result of [Criminisi 04] (9×9 patches), result of [Le Meur 11] (7×7, 3 levels of hierarchy and search window of 31 × 31), and result of the proposed GBCA method (3×4 blocks and 13×13 patches).

chosen contextually similar blocks Z is then proportional to the total number of blocks, $Z = n_b/r$, where $r = 6$, but we do not allow Z to be lower than 2. We will comment on the different choices of patch size, number of blocks and r at the end of this section. Finally, the threshold for oriented energy from Section 4.5.1 is set to $T_{OE} = 10^{-4}$. By experimental evaluation on different natural images, we determined that this value preserves the important contour features, while discarding the noisy measurements in flat areas.

As the first experiment, we compare the original priority definition from [Criminisi 04], $R_1(p) = D_c(p)D_g(p)$, where $D_c(p)$ is the confidence term and $D_g(p)$ is the gradient-based data term (see Section 4.3.3), with the proposed orientation-based priority $R(p)$ (Eq. (4.13)). For inpainting, we used the proposed context-aware patch selection with the same parameters (patch size 11×11, division into 3×3 blocks, total $n_b = 9$). Fig. 4.15 shows the inpainting results on the artificial example, where $R_1(p)$ is used in the top and $R(p)$ in the bottom row. We can clearly see that the proposed priority preserves better the structure in the image, while original priority gives preference to texture. This leads to the proposed method yielding better inpainting result.

Next, we qualitatively compare the results of the proposed method with some of the state-of-the-art patch-based methods on different natural images. Fig. 4.16 shows the comparison with the "greedy" method from [Criminisi 04][7] and a more recent non-local method from [Le Meur 11][8] (see caption for parameters). We can see that the method from [Criminisi 04] (top right) introduces artefacts because it partially duplicates the man in the green sweater. The method from [Le Meur 11] (bottom left) does not preserve well neither the image structure (because it propagates snow in the region of the sky), nor the texture. The proposed approach (bottom right) preserves well the image structure (the border between sky and snow), while introducing the least amount of artefacts in the texture of the snow. Therefore, the proposed approach outperforms both reference methods.

Better performance compared with [Criminisi 04] is also visible in Figs. 4.17 and 4.18. In the "elephant" image (Fig. 4.17), we can see that the method from [Criminisi 04] introduces artefacts, e.g., branches of the trees in the blue background area and background area in the bushes (see marked areas in the bottom left). These artefacts are not present in the result of the proposed method (bottom right). In the "vegas" image (Fig. 4.18), the method from [Criminisi 04] also introduces some artefacts, e.g., plants are not homogeneous and the socket from the lower left part of the image is inpainted into the missing area (see marked areas in the bottom left). In addition, the structure is not well-preserved. Compared with the result of the Content Aware Fill from Adobe PhotoShop, the proposed method (in the bottom right of Figs. 4.17 and 4.18) provides better continuation of structures (see marked areas in the top right).

Fig. 4.19 shows another comparison with the Content Aware Fill (top

[7] MatLab software from http://www.cc.gatech.edu/∼sooraj/inpainting/.
[8] Results were received from the authors.

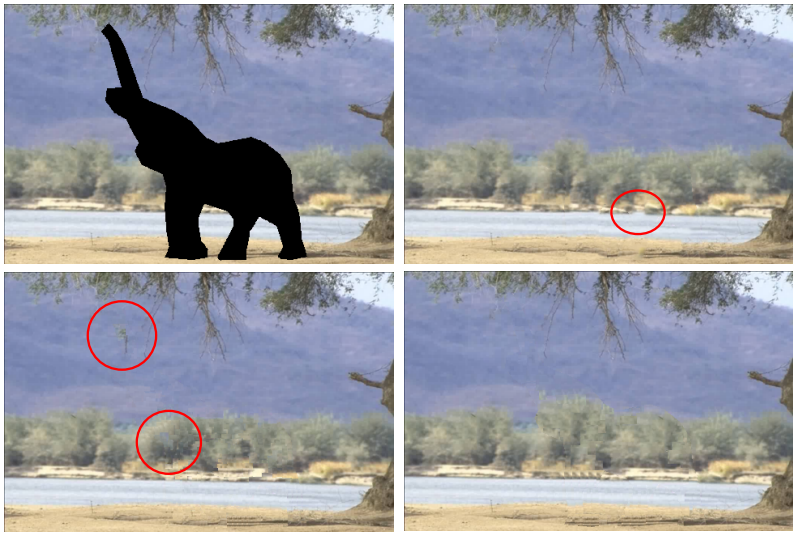


Figure 4.17: Comparison of inpainting results for the "elephant" image. From left to right and top to bottom: input image with the missing region marked in black, result of the Content Aware Fill, result of [Criminisi 04] (13×13 patches), and result of the proposed GBCA approach (5×7 blocks and 21×21 patches).

Figure 4.18: Comparison of inpainting results for the "vegas" image. From left to right and top to bottom: input image with the missing region marked in black, result of the Content Aware Fill, result of [Criminisi 04] (17×17 patches), and result of the proposed GBCA approach (4×6 blocks and 17×17 patches).

Table 4.1: Comparison of computation times for different images.

Image (patch size)	Greedy method from [Criminisi 04]	Proposed method
"baseball" (13×13)	154.64s	81.15s
"elephant" (21×21)	181.3s	88.51s
"vegas" (17×17)	155.32s	85.63s

right), and with the very recent method from [Le Meur 12][9] (bottom left). We can see that the proposed method (bottom right) outperforms the Content Aware Fill, which introduces artefacts (the man from the background is duplicated), while yielding comparable results with the method from [Le Meur 12].

Table 4.1 shows the comparison of computation times of the proposed method and the related method from [Criminisi 04] on images from Figs. 4.16, 4.17 and 4.18. The times were obtained in MatLab R2012b on Intel Core2 Quad Q9550 2.83 GHz CPU with 8GB RAM. For fair comparison in terms of parameters, we tested the algorithms for the same patch size, which was in this case the one yielding the best result of the proposed method, as indicated in the first column of the table. Note that for the "vegas" image (Fig. 4.18), the same patch size yielded the best result for both tested methods, thus the comparison of computation times for this image is fair also in terms of the quality of the result. We can see that our method is about 2 times faster than the greedy method from [Criminisi 04], which performs exhaustive search for the best-matching patch. The context-aware approach accelerates the patch search itself about 6 times, because the search space is about 6 times smaller than in the exhaustive search ($r = 6$). However, the overall speed-up is smaller due to the overhead computations, mainly updating contextual descriptors (see step 9 in Algorithm 2).

As we mentioned at the beginning of this section, our method depends on the patch size and the division into blocks of fixed size, and we showed the results for the combination of parameters that performed best. Patch size is an important parameter in all patch-based algorithms, regardless of application (texture synthesis, SR or image inpainting). It should be big enough to capture important structures in the image (in this case, the area surrounding the missing region), but not too big in order to still be able to find good matches. Therefore, for most of the inpainting methods, including ours, the choice of the "good" patch size is individual for each image and its variation influences the inpainting result. This is shown in Fig. 4.20, where inpainting results were obtained with the proposed method and the patch size 9×9, 13×13 and 17×17, with the division into 4×5 blocks and $r = 6$, thus $Z = 3$. We can see that for this image, the best result is obtained with the patch size 9×9.

The dependence on the number of blocks in the image is shown in Fig. 4.21, with the patch size fixed to 17×17 and $r = 6$. We can see that the

[9] Results were received from the authors.

Figure 4.19: Comparison of inpainting results for the "rice field" image. From left to right and top to bottom: input image with the missing region marked in black, result of the Content Aware Fill, result of [Le Meur 12], and result of the proposed GBCA approach (4×5 blocks and 15×15 patches).

Figure 4.20: Dependence of the inpainting result of the proposed method on the patch size while keeping the number of blocks fixed (4×5 blocks). From left to right and top to bottom: input image with the missing region marked in black, inpainting result with 9×9, 13×13 and 17×17 patches.

Figure 4.21: Dependence of the inpainting result of the proposed method on the division into blocks of fixed size, with 17×17 patches and $r = 6$. From left to right: input image with the missing region marked in black, inpainting result with 4×6, 5×7 and 6×9 blocks.

Figure 4.22: Dependence of the inpainting result of the proposed method on the parameter r. The results are shown for the "vegas" image (Fig. 4.18) with 17×17 patches and the division into 4×6 blocks. From left to right and top to bottom: inpainting result with $r = 3$, $r = 4$, $r = 6$ and $r = 8$.

results vary and that the best one corresponds to the division into 4×6 blocks. Furthermore, the results also depend on the fraction r, as shown in Fig. 4.22. The results get worse as the number of chosen blocks increases (i.e., r decreases) because the source region is bigger and less constrained, thus the possibility of

finding a wrong match is increased. Moreover, the computation time increases. After experimental evaluation on a number of images, we concluded that $r = 6$ gives good block-matching and final inpainting result, with a good trade-off between quality and speed.

The disadvantage of the proposed method is the number of parameters that needs to be set in order to obtain the inpainting result. In Chapter 5, we will make an attempt to diminish the number of parameters. In this chapter, our goal was to illustrate the proof of concept for context-aware patch selection and to introduce a novel patch priority. We also showed that such an approach outperforms in terms of both quality and speed the method from [Criminisi 04], and often even some state-of-the-art methods (in terms of quality).

4.7 Conclusion

In this chapter, we introduced the image inpainting problem, where the goal is to fill in the missing or damaged part of the image by using the known (undamaged) part. This region to be inpainted is assumed to be known, either from the user-provided information or as an output of some automatic detection. We made an overview of different image inpainting methods, with the focus on patch-based algorithms, which we viewed from three different aspects: patch selection, patch search and patch priority.

The main contribution of this chapter is a novel context-aware patch-selection approach, which reduces the number of candidate patches and chooses them in such a way that they better fit the surrounding context. Context is represented within blocks of fixed size using contextual descriptors in the form of combined texture and colour features. Comparison of these contextual descriptors enables us to find regions of similar context in the image, as we demonstrated with intermediate results. Such context-aware approach is general and thus can be applied within any patch-based inpainting algorithm. In this chapter, we applied it within a novel greedy inpainting algorithm, called greedy block-based context-aware (GBCA) method, where we proposed a novel orientation-based priority. This definition of priority is based on contour features, which are obtained by filtering the image with the bank of Gabor filters at multiple scales and orientations. The same approach was applied for the extraction of texture features used for context representation. The results obtained with the proposed method illustrate the potential of our approach. We demonstrated that the meaningful constrained search for patches yields better inpainting result in less time than the exhaustive search. Furthermore, our results are visually better or comparable with state-of-the-art methods.

Parts of this work resulted in several conference publications [Ružić 12a, Ružić 12b, Ružić 13a], while the method was fully presented in [Ružić 13c].

This chapter also serves as an introduction to the remaining of this thesis, since it gives a thorough overview of patch-based inpainting methods and introduces some preliminary ideas about context-aware approach. In the next

chapter, those preliminary ideas will evolve into a solid framework resulting from more elaborate analysis and validation and from improvements on various aspects of contextual descriptors, block division strategy and patch-selection approach.

5

MRF-based image inpainting with context-aware label selection

In this chapter, we further develop our context-aware approach for patch-based inpainting, introduced in Chapter 4. The main contributions of this chapter are a novel MRF-based inpainting method, which uses the context-aware approach, and improved context representation.

At the beginning of the chapter, in Section 5.1, we introduce a general approach for MRF inpainting, together with an efficient optimization method called priority belief propagation from [Komodakis 07]. Next, in Section 5.2 we propose the improved context representation compared to the one we proposed in Section 4.4.2. First of all, we propose to use texton histograms as contextual descriptors, which prove to be more effective than averaged filter outputs and averaged colour we introduced in the previous chapter. Second, we describe the context within blocks of *adaptive* sizes. Finally, the division of the image into blocks of adaptive sizes is obtained with the novel top-down splitting procedure, which we describe in Section 5.2.2.

A novel MRF-based approach with the context-aware label selection is introduced in Section 5.3. Another important contribution of the proposed approach is a novel optimization approach, which extends our inference method from Section 2.4 in order to deal with the problems with huge number of labels. The results, presented in Section 5.4, demonstrate potential of the proposed inpainting method for two applications: scratch and text removal and object removal.

5.1 MRF-based image inpainting

A promising approach for patch-based image inpainting is to treat inpainting as a global optimization problem (see Section 4.3.1.3 for an overview of global inpainting methods). This approach allows multiple candidate patches (labels) to eventually choose one label for each position so that the whole set of labels (at all positions) minimizes a global optimization function. Among different global methods, MRF-based methods [Komodakis 07, Huang 07, Yang 09] combine patch-based models and the MRF framework, which models the global image context via local interactions (see Chapter 2). In particular in this section, we visit in more detail the seminal work of [Komodakis 07], which is the starting point for our method, proposed later in Section 5.3.

5.1.1 Notations and definitions

Patch-based image inpainting via MRF modelling proposed in [Komodakis 07], assumes an MRF model (see Fig. 2.3) over a target region Ω of a damaged input image g. The MRF lattice S consists of pixel positions, which are w pixels apart in horizontal or vertical direction on the image lattice I, and where $(2w+1) \times (2w+1)$ masks $\Psi + i$ centred at the positions $i \in S$ intersect the target region, i.e., $(\Psi + i) \cap \Omega \neq \emptyset$ (see Fig. 5.1 for graphical representation and Section 3.3 for the definition of the mask). The positions i on the lattice $S \subset I$ thus represent MRF nodes. The first-order neighbourhood system is considered with pairwise cliques $\langle i, j \rangle$. Note that the masks centred at neighbouring nodes are overlapping.

Rather than estimating a single pixel at each node i, all pixels within a $(2w+1) \times (2w+1)$ mask $\Psi + i$ are estimated at once. Therefore, the values, i.e., the labels, which are to be assigned to nodes are patches. In particular, labels are all possible $(2w+1) \times (2w+1)$ patches from the input image that are completely inside the source region Φ, i.e., that have *no* missing pixels (see Fig. 5.1). According to notations introduced in Section 3.3, an image patch (thus also a label) is specified by its central pixel position, image g, and mask shape Ψ. Since in this application all labels come from the same image and are specified with the same mask, we refer to labels only by their central pixel positions. Now the label position set can be formally defined as

$$\Lambda = \{p \in I | (\Psi + p) \subset \Phi\}. \tag{5.1}$$

The assignment of a label to the node i amounts to copying the values from the patch centred at $x_i \in \Lambda$ to the positions within the mask $\Psi + i$ centred at the node i in the image. An illustration of the MRF notations introduced above is shown in Fig. 5.2.

The Bayesian estimator requires specifying both the prior (MRF model) and the likelihood model, i.e., the data cost (see Section 2.2.4 for more details). The data cost models the relationship between the label centred at x_i and the observation y_i at each node i (see Fig. 2.3 and Section 2.2.4). In this

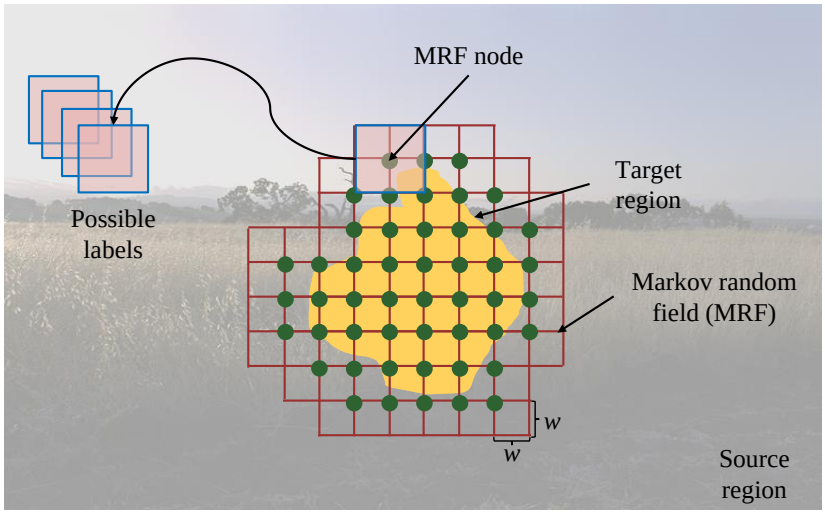


Figure 5.1: Illustration of the MRF for inpainting.

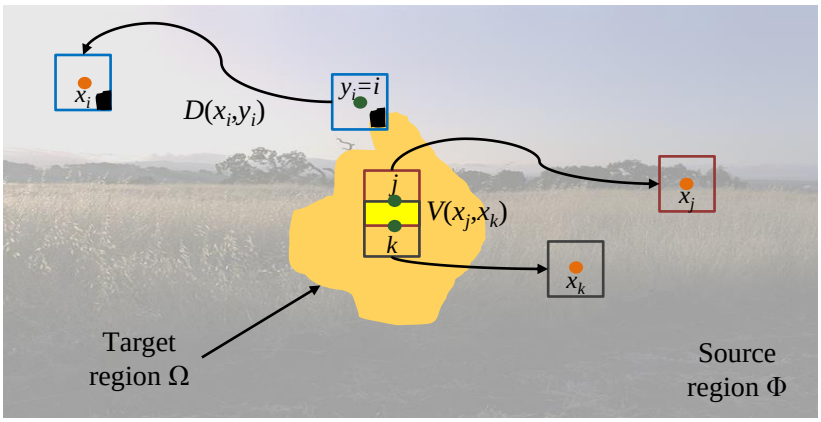


Figure 5.2: Illustration of the MRF notations. The green circles, corresponding to positions $i, j, k \in S$, denote MRF nodes, and the orange circles denote the central positions $x_i, x_j, x_k \in \Lambda$ of their corresponding labels, which are the whole patches of pixel values centred at these positions. The black areas in the patches centred at i (or equivalently y_i) and x_i mark the locations of missing pixels at the node i. The data cost $D(x_i, y_i)$ is computed over the non-black areas of these patches. The pairwise potential $V(x_j, x_k)$ is computed over the light yellow region.

patch-based MRF model for inpainting, the observation, i.e., the available information at the node i, is the existing image patch centred at i, whose content

is partially or fully unknown. Since we refer to image patches by their central positions, y_i denotes the central position of the observation at node i, resulting in $y_i = i$ (see also Fig. 5.2).

The data cost $D(x_i, y_i)$ measures the agreement of the label centred at x_i with the known pixels of the image patch (observation) centred at y_i, thus it is defined for each label as

$$D(x_i, y_i) = \begin{cases} \|\mathcal{T}_{x_i} g - \mathcal{T}_{y_i} g\|^2_{(\Phi - y_i) \cap \Psi}, & \text{if} \quad (\Phi - y_i) \cap \Psi \neq \emptyset \\ 0, & \text{if} \quad (\Phi - y_i) \cap \Psi = \emptyset, \end{cases} \tag{5.2}$$

where the translation operator $\mathcal{T}$ and the norm are defined in Section 3.3. Therefore, if there is a known part of the observation centred at y_i, the data cost is the SSD between that known part and the corresponding part of the label centred at x_i. Otherwise, if the observation is completely inside the missing region Ω, the data cost is zero. The nodes whose observations are completely inside the missing region are called interior nodes. Finally, the pairwise potential $V(x_i, x_j)$ is similarly defined as the SSD between labels centred at x_i and x_j in their nodes' region of overlap

$$V(x_i, x_j) = \|\mathcal{T}_{x_i} g - \mathcal{T}_{x_j + (i-j)} g\|^2_{\Psi \cap (\Psi - (i-j))}, \tag{5.3}$$

(see also Fig. 3.13). The global inpainting problem can now be formulated as minimizing the energy

$$E(\mathbf{x}|\mathbf{y}) = \sum_{\langle i,j \rangle} V(x_i, x_j) + \sum_i D(x_i, y_i). \tag{5.4}$$

As a result of this minimization, one label is chosen per MRF node so that all labels (over all nodes) agree with each other as much as possible globally.

Note that this MRF model has some similarities with the MRF model used for patch-based super-resolution (SR) [Freeman 00, Ružić 11b], which we also used in our SR method (Section 3.4). In particular, the MRF model for SR is also defined on a lattice of nodes, which represent the central positions of overlapping square masks. The pixel values within these masks are to be estimated by the MRF. The data cost, which is in SR actually defined in probability form via local evidence (see Eq. (3.15)), also measures the agreement between the available information at the node and its labels, while the pairwise potential (or equivalently pairwise compatibility in Eq. (3.16)) measures the agreement of labels in their nodes' region of overlap. However, there are several differences. First of all, for image inpainting the lattice of nodes covers *only* the target region and not the whole image. Second, the observation represents the actual image patch at the position of the node in the image, while in SR that is the low-resolution (LR) patch at the corresponding position in the LR image. Consequently, the data cost is computed in a different way. Finally, labels have different interpretation (patches from the image at the same resolution instead of high-resolution patches).

Minimizing the energy from Eq. (5.4) could be solved using loopy belief propagation (LBP) algorithms [Yedidia 01b] (see also Section 2.3.5), like in the SR application in [Freeman 00]. However, applying LBP directly (or any standard inference method) may be prohibitive due to the huge number of labels of each node. The inpainting method of [Komodakis 07] introduced an improved version of belief propagation called priority belief propagation (p-BP), to deal more efficiently with such problems.

5.1.2 Priority belief propagation

The main goal of p-BP [Komodakis 07] is to reduce the number of labels in a meaningful way, which leads to an efficient minimization of Eq. (5.4). This is achieved by adding two extensions to LBP: *priority message scheduling* and *label pruning*. Therefore, the core of the algorithm is still LBP, because information is propagated throughout the graph by communicating messages between the nodes (Eq. (2.25), Fig. 2.5(a)), while label assignment is based on the value of belief (Eq. (2.26), Fig. 2.5(b)). Message and belief can be defined in log domain as

$$m_{ij}(x_j) = \min_{x_i \in \Lambda} \{V(x_i, x_j) + D(x_i, y_i) + \sum_{k \in \partial i : k \neq j} m_{ki}(x_i)\} \tag{5.5}$$

$$b_i(x_i) = -D(x_i, y_i) - \sum_{k \in \partial i} m_{ki}(x_i), \tag{5.6}$$

respectively, where ∂i is the neighbourhood of the node i. Belief describes how likely is that the label centred at x_i will be assigned to the node i. The maximization of beliefs, which takes place after the algorithm has converged, leads to the maximum a posteriori (MAP) estimate.

Considering the huge number of labels in the inpainting application (tens of thousands, depending on the image size), it is obvious that messages are very expensive to compute. Therefore, the important extension introduced by the p-BP algorithm is label pruning. Its purpose is to reduce the number of possible labels for each node to some number $L \in [L_{min}, L_{max}]$, where $L_{max} \ll \#\Lambda$. This is achieved by discarding unlikely labels, i.e., the labels whose relative belief $b_i^{rel}(x_i) = b_i(x_i) - b_i^{max}$ (where $b_i^{max} = \max_{x_i \in \Lambda} b_i(x_i)$) is smaller than some threshold b_{prune} (for details, see [Komodakis 07]). However, beliefs of interior nodes are zero for all labels, because the data cost is zero (see Eq. (5.2)), thus the node cannot know which labels to prefer.

In order to solve this problem, priority message scheduling was introduced as another extension to LBP in the following manner. Previously unvisited nodes are visited in the order of the highest priority, their labels are pruned, messages are sent to their unvisited neighbours, and beliefs and priorities of those neighbours are updated. This means that label pruning is performed for the node of the highest priority, and priority is defined in such a way to ensure that the node has sufficient information about which labels

Algorithm 3 Priority belief propagation (p-BP)

assign priorities to nodes and declare them as "unvisited"
for $t = 1$ **to** N_{iter} **do** {N_{iter} is the total number of iterations}
    forward pass:
    for $n = 1$ **to** N_n **do** {N_n is the total number of nodes}
        i = "unvisited" node of the highest priority
        apply label pruning to node i
        order[n] = i
        declare i as "visited"
        for any "unvisited" neighbour $j \in \partial i$ **do**
            send all messages $m_{ij}(x_j)$ from node i to node j, $\forall x_j \in \Lambda$
            update belief $b_j(x_j)$, $\forall x_j \in \Lambda$, and priority $R(j)$ of node j
        end for
    end for
    backward pass:
    for $n = N_n$ **to** 1 **do**
        i=order[n]
        declare i as "unvisited"
        for any "visited" neighbour $j \in \partial i$ **do**
            send all messages $m_{ij}(x_j)$ from node i to node j, $\forall x_j \in \Lambda$
            update belief $b_j(x_j)$, $\forall x_j \in \Lambda$, and priority $R(j)$ of node j
        end for
    end for
end for
assign $\hat{x}_i = \arg\max_{x_i \in \Lambda} b_i(x_i)$, $\forall i \in S$

to prefer, i.e., that is confident about its labels. In particular, priority $R(i)$ is inversely proportional to the number of labels whose relative belief $b_i^{rel}(x_i)$ is equal to or higher than some threshold b_{conf}:

$$R(i) = \frac{1}{\#\{x_i \in \Lambda | b_i^{rel}(x_i) \geq b_{conf}\}}. \tag{5.7}$$

This means that the nodes with more confidence about their labels will have higher priority. In practice, those are the nodes lying on an image structure and having more known pixels. The benefit of priority message scheduling is twofold: it makes label pruning possible, thus allowing "cheap" computation of messages, and it makes the inference algorithm converge faster [Komodakis 07].

The above described algorithm represents only one part of p-BP, called the forward pass. The other part is the backward pass, where nodes are visited in the reverse order and the rest of the messages are sent and the beliefs and priorities are updated. The forward and the backward pass are conducted over multiple iterations (see pseudo-code in Algorithm 3). Note that label pruning does not take place in the backward pass. In fact, in practice it takes place only during the *first* forward pass, because after that L labels will be

chosen for each node, and it will not be necessary to perform label pruning again.

It is also important to notice that both label pruning and priority computation are based on beliefs. Since beliefs are computed during the inference algorithm, it means that the information necessary for algorithm's efficiency is obtained from the algorithm itself. However, a problem with p-BP is that, prior to label pruning, all possible labels for each node are considered. Therefore, the message and belief computations in the first forward pass of the algorithm (steps 10 and 11 in Algorithm 3) are performed for a huge number of variables, which makes the algorithm very slow, especially when applied on bigger images.

5.2 Improved context representation

In Section 4.4, we proposed a general approach for context-aware inpainting, where the main idea is to constrain the patch search to image areas that are contextually similar to the area surrounding the missing region. This idea is illustrated in Fig. 4.8. We proposed context representation, which is based on image division into square non-overlapping blocks of fixed size, where to each block $B + l$, $l \in \Theta$, a contextual descriptor $\mathbf{c}^{(l)}$ is assigned. This descriptor consists of texture and colour features (Eqs. (4.1) and (4.2)), where texture features are extracted by averaging magnitudes of filter responses obtained by filtering the image with the bank of Gabor filters.

In this section, we aim at improving context representation in two ways. First, we explore the use of normalized texton histograms [Leung 99, Malik 99, Malik 01] as contextual descriptors (see Section A.5 for more details on textons and texton histograms). Texton histograms were previously used in [Leung 99, Leung 01, Varma 02, Cula 04, Varma 05] for texture classification and recognition, and in [Malik 01, Arbelaez 11] to estimate the texturedness of a pixel for the purpose of image segmentation. However, to our knowledge, their use for image inpainting application has never been explored before.

Another improvement that we introduce in this section is the more sophisticated division of the image into blocks of adaptive sizes. In order to obtain this division, we propose a novel *top-down splitting procedure*, which is also based on contextual descriptors (Section 5.2.2).

5.2.1 Texton histograms as contextual descriptors

In this chapter, we will use textons as they were originally defined for grey-scale images in [Malik 99, Malik 01] (some alternative definitions and interpretations are reviewed in Section A.5.1). This original approach includes filtering the image with a bank of oriented multi-scale filters, followed by K-means clustering of normalized filter responses. Textons are then defined as the K cluster centres, each being a vector of dimensionality equal to the total number of filters in the filter bank. Each pixel is mapped to the texton that is the closest to its

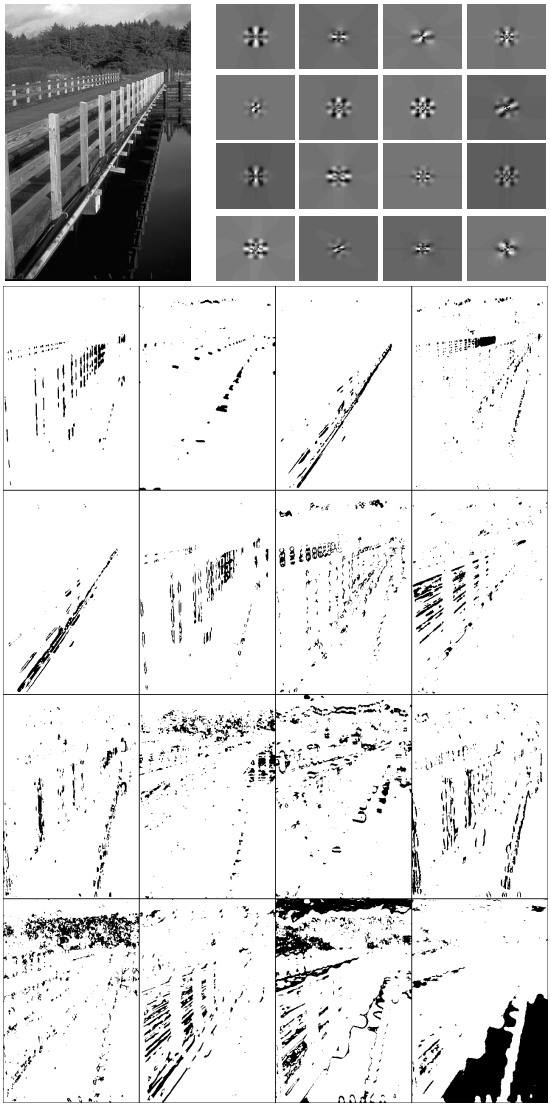


Figure 5.3: Top left: input image. Top right: textons found via K-means clustering of magnitudes of complex Gabor filter outputs ($K = 16$), sorted in raster-scan order by decreasing norm. Bottom: mapping of pixels to each texton channel.

vector of filter responses (in terms of a distance between two vectors), resulting in a pixel-to-texton mapping T (see Section A.5 for more details). In our approach to computing textons, we use the bank of complex Gabor filters (see Appendix A), and we cluster magnitudes of complex filter responses. For the image in the top left of Fig. 5.3, we obtained the textons shown in the top right of Fig. 5.3. The textons are visualized by pre-multiplying each vector

corresponding to the texton by the pseudo-inverse of the filter bank, as proposed in [Jones 92]. The bottom of Fig. 5.3 represents pixel-to-texton mapping T, where each binary image corresponds to one texton. We can see that textons correspond to oriented edge elements (most of the textons in the first three rows), texture (e.g., the first three textons in the last row) and smooth areas (the last texton in the last row).

Now we can easily substitute averaged filter responses that we used in Chapter 4 with texton-based analysis, with the goal of improving context description. In particular, we propose to use normalized texton histograms, similar to those in [Leung 99, Malik 99, Malik 01] (see Section A.5 for more details), as contextual descriptors for each image block. Therefore, we define the contextual descriptor $\mathbf{c}^{(l)}$ of the block $B + l$ as the normalized texton histogram, where each bin of the histogram represents one element of the vector $\mathbf{c}^{(l)}$:

$$c_n^{(l)} = \frac{1}{\#((B+l) \cap \Phi)} \sum_{p \in ((B+l) \cap \Phi)} \xi[T(p) = n], \quad n = 1, \ldots, K. \tag{5.8}$$

Therefore, only known pixels from $B + l$ are used for computation because we have to consider that some pixels are missing. ξ is the indicator function, i.e., it returns one if its argument is true and zero otherwise.

To compare blocks by their context, we introduced in Eq. (4.4) a general dissimilarity measure $\bar{H}^{(l,m)}$ between blocks $B+l$ and $B+m$ as some distance between their corresponding contextual descriptors $\mathbf{c}^{(l)}$ and $\mathbf{c}^{(m)}$. Since contextual descriptors are now defined as texton histograms, this general dissimilarity measure is now specified as a common χ^2 test:

$$\bar{H}^{(l,m)} = \chi^2(\mathbf{c}^{(l)}, \mathbf{c}^{(m)}) = \frac{1}{2} \sum_{n=1}^{K} \frac{(c_n^{(l)} - c_n^{(m)})^2}{c_n^{(l)} + c_n^{(m)}}. \tag{5.9}$$

An illustration of different image blocks, their texton histograms and histogram dissimilarity is shown in Fig. 5.4. We can see that the context similarity of image regions is reflected in the texton histogram dissimilarity measure: the dissimilarity is much lower between two regions consisting of mainly flat areas, $\chi^2(\mathbf{c}^{(l)}, \mathbf{c}^{(m)})$, and high between a flat and a textured region, $\chi^2(\mathbf{c}^{(l)}, \mathbf{c}^{(n)})$.

In our experiments, using texton histograms as defined above requires less parameters and easier parameter optimization than our earlier contextual descriptors from Chapter 4. This will be further discussed in Section 5.4.3.

5.2.2 Image division into blocks of adaptive sizes

So far, we considered a simple image division into fixed-size blocks, where contextual descriptors are computed as some statistics of texture features within the blocks. However, if a block is inhomogeneous, i.e., it contains different textures, that statistics does not necessarily represent well any of the textures

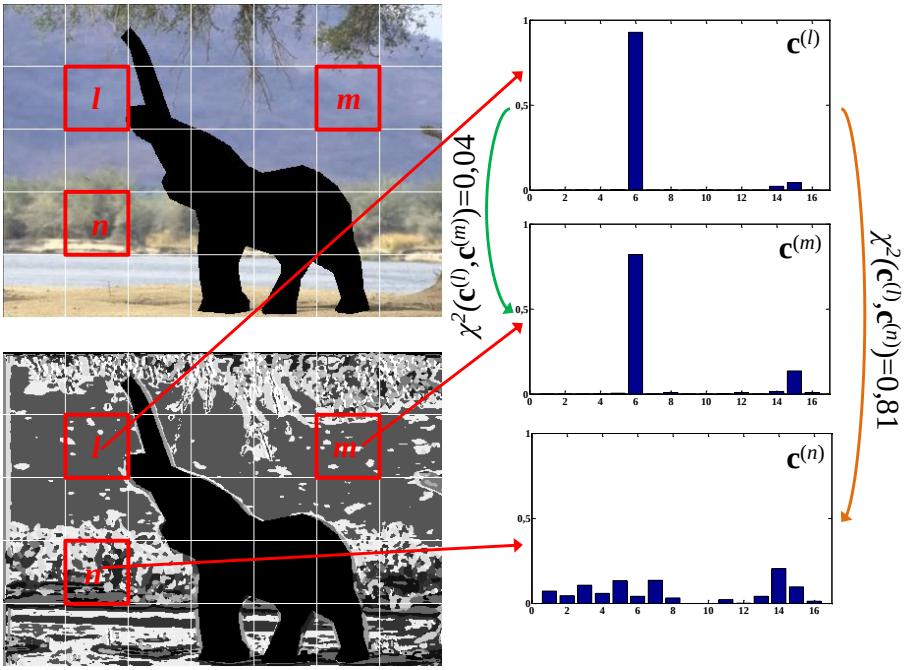
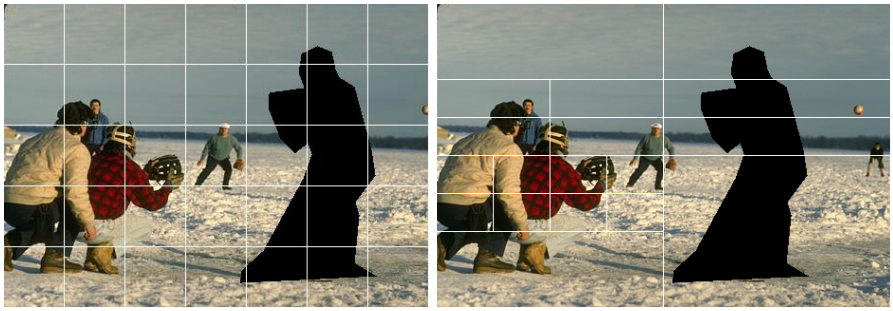


Figure 5.4: Top left: original image with the missing region marked in black. Bottom left: pixel-to-texton mapping for all textons together with three different image blocks marked with red squares (l, m, n are the central positions of the blocks). Right: normalized texton histograms as contextual descriptors corresponding to the marked blocks, and their histogram dissimilarity.



(a) **(b)**

Figure 5.5: Division into blocks: (a) division into 5×7 blocks of fixed size and (b) division into blocks of adaptive sizes.

present in a block. Therefore, in most natural images, some image areas call for finer division than the others (see the example in Fig. 5.5 and see how the

number of blocks of fixed size influences the inpainting result in Fig. 4.21). Moreover, the optimal size of blocks can differ from one image to another (see the results in Section 4.6). We define a novel top-down splitting procedure that automatically divides the image into blocks of adaptive sizes. The idea is to start from some coarse division of the input image into blocks, which are further divided depending on the "homogeneity" of their texture. This homogeneity is determined by the statistical test, which measures the similarity of texture features.

Fig. 5.6 illustrates the proposed splitting procedure for one image block B. We need to favour that the splits in horizontal and vertical direction alternate through levels in order to prevent splitting along one direction only. Therefore, we assign each block a binary variable $\delta \in \{h, v\}$, that we call *directional flag*. This variable determines the direction, horizontal (h) or vertical (v), along which the evaluation of the block's homogeneity will have the priority. Let B_1^d and B_2^d denote two sub-blocks of B along direction $d \in \{h, v\}$, and $\mathbf{c}_d^{(1)}$ and $\mathbf{c}_d^{(2)}$ the corresponding texton histograms. We define the measure of inhomogeneity of the block B along direction d as the χ^2 dissimilarity measure from Eq. (5.9):

$$\bar{H}_d = \bar{H}_d^{(1,2)} = \chi^2(\mathbf{c}_d^{(1)}, \mathbf{c}_d^{(2)}), \quad d = h, v, \tag{5.10}$$

and we define the *split variable* s_d along direction d as:

$$s_d = \begin{cases} 1, & \text{if} \quad \bar{H}_d > T_b \\ 0, & \text{if} \quad \bar{H}_d \leq T_b, \end{cases} \tag{5.11}$$

where T_b is the block similarity threshold. If the block B is inhomogeneous along direction d, then the value of s_d signals splitting along that direction. In practice, we allow splitting only along one direction at the time. Therefore, we initialize s_h and s_v to zero and we evaluate them sequentially, in the order that depends on the directional flag δ. If $\delta = h$, we first evaluate s_h, and then *only* if $s_h = 0$, we evaluate s_v. The order is reversed when $\delta = v$. Hence, there are only three possible outcomes for (s_h, s_v) in our algorithm: $(0, 0)$ implies no further splitting of B, $(0, 1)$ implies splitting vertically and $(1, 0)$ splitting horizontally.

If the block B is split along one of the directions, each of the two new sub-blocks B_j, $j = 1, 2$, can be declared as amenable to further splitting or not, depending on the reliability check, denoted as $Q^{(j)}$ in the algorithm in Fig. 5.6. If the size of the block B_j is above a certain fraction r of the input image dimensions, and if $\rho^{(j)} = 1$, where $\rho^{(j)}$ is the reliability from Eq. (4.7), then $Q^{(j)} = 1$. This means that the block B_j is allowed to be tested for further splitting, and its directional flag $\delta^{(j)}$ is set to the direction opposite of the split by which the sub-block was generated. If $Q^{(j)} = 0$, the block B_j may not be split any further. This reliability check prevents already unreliable and/or too small block from being divided into less meaningful parts.

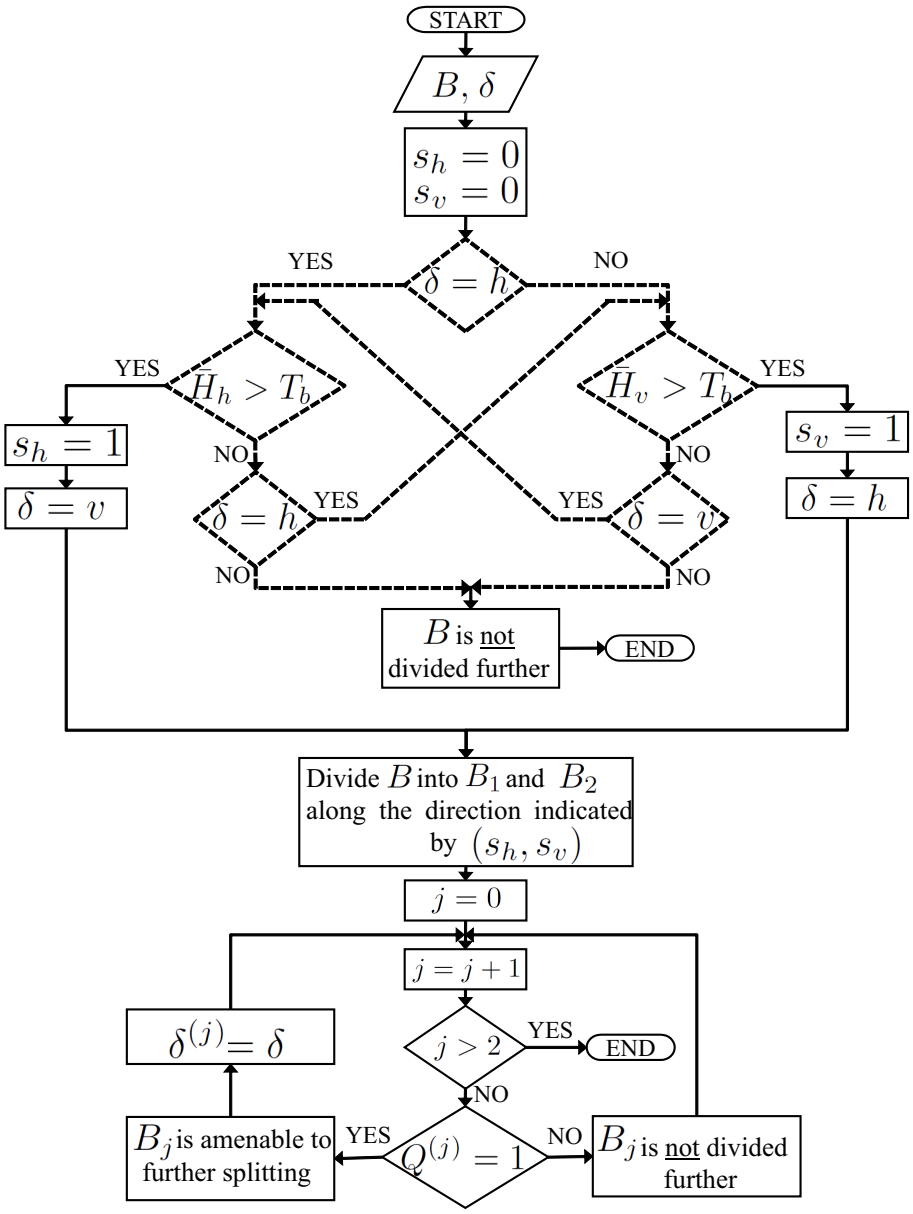


Figure 5.6: Block diagram of the proposed top-down splitting procedure (see text for notations). The core of the algorithm is indicated with dashed lines.

The above described algorithm, illustrated in Fig. 5.6, can be applied on any image block that is amenable to further splitting. We start the complete top-down splitting procedure from the initial coarse division of the image into four blocks, and we apply the above described algorithm on each of these blocks and all of their sub-blocks (amenable to further splitting), at all the levels. The top-down splitting procedure finishes once there are no more blocks that are allowed to be tested for further splitting. The output is the division of the image into blocks of adaptive sizes (see Fig. 5.5b), which consists of all the blocks, from all the levels, which were not divided further. According to the previously introduced notations, we will refer to these blocks by their central positions $l \in \Theta$, thus the block is denoted as $B+l$. The set Θ is now determined by the adaptive division. Note that the sizes of these blocks differ, but the size does not play a part in our equations. Therefore, we will not express it explicitly, but rather assume that it is determined automatically by the central position of the block.

5.3 MRF block-based context-aware (MBCA) inpainting method

In this section, we propose a novel context-aware MRF-based inpainting algorithm, where the search for labels is constrained to the regions of well-matching context. For this purpose, we adapt our general approach for context-aware patch (i.e., label) selection, presented earlier in Section 4.4, to the improved context representation introduced in Section 5.2. Specifically, this modification accounts for a new type of block division, new contextual descriptors and a new criterion for choosing contextually similar blocks, as we will explain in detail in Section 5.3.1. We call the proposed method MRF block-based context-aware (MBCA) method.

We encode prior knowledge about the spatial consistency between neighbouring image patches using an MRF model, similarly as in [Komodakis 07] (see Section 5.1.1 for notations and definitions). Although we limit the label set with our context-aware approach, the number of labels is still too big and most of the existing inference methods will be inefficient. Therefore, we propose a novel optimization approach in Section 5.3.2, which is suitable for global inpainting problem with large number of labels.

5.3.1 Context-aware label selection

Regardless of the division strategy (into blocks of fixed or adaptive sizes), the idea of our context-aware approach is to constrain the source region for unknown patches, belonging to some current block $B + l$, to a region $\Phi^{(l)} \subset \Phi$ with the context well matching that of $B + l$. Recall from Section 4.4.3 that $\Sigma^{(l)}$ denotes the set of indices of the blocks that are contextually similar to the current block $B + l$. If we use texton histograms $\mathbf{c}^{(l)}$ from Eq. (5.8) as contextual descriptors, then the set $\Sigma^{(l)}$ is determined as

$$\Sigma^{(l)} = \{m | \bar{H}^{(l,m)} \leq T_b\}, \tag{5.12}$$

where $\bar{H}^{(l,m)}$ is defined as in Eq. (5.9), and T_b is the block similarity threshold from Section 5.2.2. Then $\Phi^{(l)}$ is defined as

$$\Phi^{(l)} = \begin{cases} \cup_{m \in \Sigma^{(l)}} ((B+m) \cap \Phi), & \text{if} \quad \rho^{(l)} = 1 \\ ((B+l) \cap \Phi) \cup \big(\cup_{\substack{l' \in \partial l \\ \rho^{(l')}=1}} \Phi^{(l')} \big) \cup \big(\cup_{\substack{l' \in \partial l \\ \rho^{(l')}=0}} ((B+l') \cap \Phi) \big), & \text{if} \quad \rho^{(l)} = 0, \end{cases} \tag{5.13}$$

taking into account both reliable and unreliable blocks, since this distinction can also be made for blocks of adaptive sizes (see Eq. (4.7) for the definition of reliability). Note that the constrained source region $\Phi^{(l)}$ of the current *unreliable* block $B+l$ according to Eq. (5.13) is different from the one defined earlier in Eq. (4.8). Here, it additionally includes neighbouring unreliable blocks themselves ($\cup_{\substack{l' \in \partial l \\ \rho^{(l')}=0}} ((B+l') \cap \Phi)$). This is because in the MRF-based approach, $\Phi^{(l)}$ remains fixed throughout the algorithm, i.e., it is determined once at the beginning, as opposed to the "greedy" approach proposed in Section 4.5.2, where it is re-evaluated in each iteration. This also means that unreliable blocks stay unreliable while performing label selection, thus we have to make sure that at the start we include all contextually similar regions of the image. Note that the pseudo-code in Algorithm 1 still applies, just that now $\bar{H}^{(l,m)}$ and $\mathbf{c}^{(l)}$ are differently defined and blocks are of adaptive sizes.

Now we can apply the above described approach for the context-aware label selection in MRF-based inpainting. According to the notations introduced in Section 4.4.1, $B + \zeta(i)$ denotes the block that contains the node i. As a result of context-aware label selection, the labels of i are all possible patches that are completely inside $\Phi^{(\zeta(i))}$. This results in a node-specific label position set, which we can formally define as

$$\Lambda_i = \{p \in I | (\Psi + p) \subset \Phi^{(\zeta(i))}\}. \tag{5.14}$$

Therefore, the node i can take a label centred at $x_i \in \Lambda_i$, where $\#\Lambda_i < \#\Lambda$ (Λ was defined in Eq. (5.1)).

5.3.2 Efficient energy minimization

We propose an efficient inference method, which builds on our general inference approach neighbourhood-consensus message passing (NCMP), introduced earlier in Section 2.4, in order to deal with MRF problems with a large number of labels. This approach shares some ideas about label pruning and priority scheduling from p-BP [Komodakis 07] (see also Section 5.1.2), in the sense

Algorithm 4 Efficient energy minimization

initialization:
for $i = 1$ **to** N_n **do** {N_n is the total number of nodes}
    compute $D(x_i, y_i)$ (Eq. (5.2))
    compute priority $R(i)$ (Eq. (5.17))
    set $\nu_i = 0$ {indicates whether the node is unvisited ($\nu_i = 0$) or visited ($\nu_i = 1$)}
end for
label pruning:
compute $D^W(x_i, y_i)$, $\forall i \in S$ (Eq. (5.18))
for $n = 1$ **to** N_n **do**
    $\hat{i} = \arg\max_{i:\nu_i=0} R(i)$
    apply label pruning: choose $L \ll \#\Lambda_{\hat{i}}$ labels centred at $x_{\hat{i}}$ that yield L smallest $D^W(x_{\hat{i}}, y_{\hat{i}})$
    for any $j \in \partial\hat{i}$ such that $\nu_j = 0$ **do**
        update $D(x_j, y_j)$ (Eq. (5.19)), $D^W(x_j, y_j)$ and $R(j)$
    end for
    set $\nu_{\hat{i}} = 1$
end for
inference method: $\hat{\mathbf{x}} = \arg\min E(\mathbf{x}|\mathbf{y})$

that we visit the nodes in some meaningful order and discard unnecessary labels. However, our approach differs on several major points. Firstly, we apply context-aware label selection to limit the number of labels. Secondly, we apply priority scheduling and label pruning only *once* and prior to the actual inference, while in p-BP these two steps are a part of the message-passing process (which was expensive in terms of memory and computational effort and even prohibitive for application on bigger images). Thirdly, we introduce new formulations of priority scheduling and label pruning, whose computation is simpler and more memory efficient than computations in p-BP. Finally, we employ a different message-passing inference algorithm to obtain the final inpainting result.

We divide the optimization process into three steps: initialization (computing priorities of nodes), label pruning (based on nodes' priorities), and the actual inference. The pseudo-code of the proposed efficient energy optimization is given in Algorithm 4.

5.3.2.1 Initialization

This step assigns priorities to all MRF nodes, which determine their visiting order in the next phase (label pruning). Like in p-BP, we shall assign higher priority to nodes that are more confident about their labels. Since in our case the number of labels $\#\Lambda_i$ for each node i can be different, we define the priority in terms of the *relative* number of confident labels RNC_i as

$$R(i) = \frac{1}{RNC_i}. \tag{5.15}$$

Our idea is to determine this RNC_i without the need to compute beliefs, but rather based on the data cost $D(x_i, y_i)$ defined in Eq. (5.2). To this end, let us define the *relative* data cost between the label centred at x_i and the observation centred at y_i as

$$D^{rel}(x_i, y_i) = D(x_i, y_i) - \min_{x_i \in \Lambda_i} D(x_i, y_i). \tag{5.16}$$

Now we determine RNC_i and the corresponding priority $R(i)$ as

$$R(i) = (RNC_i)^{-1} = \Big(\frac{1}{\#\Lambda_i} \sum_{x_i \in \Lambda_i} \big(T_R - D^{rel}(x_i, y_i)\big)_+ \Big)^{-1} \tag{5.17}$$

where $(\tau)_+ = 1$ if $\tau > 0$ and zero otherwise, and T_R is the threshold for the relative data cost, under which the assignment of a label to a node is considered as confident. Practical computation of this parameter is explained in Section 5.4. According to this priority definition, interior nodes, i.e., the nodes whose observations have no known pixels, have the lowest priority, because by definition their data cost is zero. Finally, we assign each node a binary variable ν_i, which indicates whether the node has been visited ($\nu_i = 1$) or not ($\nu_i = 0$). Initially, $\nu_i = 0, \forall i \in S$.

5.3.2.2 Label pruning

This step reduces the number of labels at each node i to a relatively small number $L \ll \#\Lambda_i$ of the "best" candidate labels. To decide which labels are the best candidates, we need a suitable distance measure. This distance measure needs to take into account:

- data fidelity, as the agreement between the undamaged part of the observation centred at y_i and the corresponding part at the label centred at x_i,
- contextual similarity between the regions (blocks) $B+\zeta(y_i)$ and $B+\zeta(x_i)$ that contain y_i and x_i, respectively.

One such possible label-pruning distance measure is contextually-weighted data cost that we define as

$$D^W(x_i, y_i) = \left(1 - e^{-\bar{H}^{(\zeta(x_i), \zeta(y_i))} - T_b}\right) D(x_i, y_i), \tag{5.18}$$

where $D(x_i, y_i)$ is the data cost and $\bar{H}^{(\zeta(x_i), \zeta(y_i))}$ is the contextual dissimilarity between image blocks containing x_i and y_i, defined in Eq. (5.9). T_b is the block similarity threshold from Section 5.2.2. Note that the weighting factor

$1 - e^{-\bar{H}^{(\zeta(x_i),\zeta(y_i))} - T_b}$ becomes very small when image blocks containing x_i and y_i are contextually similar, and tends to one when the contextual dissimilarity is very large. The constant T_b in the exponent prevents that the weighting factor becomes zero when $\bar{H}^{(\zeta(x_i),\zeta(y_i))} = 0$, and enables in this way that the labels centred at x_i coming from the contextually ideally matching region can still be ordered based on their data cost $D(x_i, y_i)$.

After computing the label-pruning distance measure $D^W(x_i, y_i)$ for each node, the nodes are visited in the order of their priority (Eq. (5.17)), keeping L labels with the smallest $D^W(x_i, y_i)$ and discarding the rest. When one node chooses its labels, this information can be propagated to its neighbouring nodes. In this way, those neighbouring nodes have more information based on which they can perform label pruning, while also the agreement of labels of neighbouring nodes can be enforced. As mentioned earlier, the data cost $D(x_i, y_i)$ of interior nodes and consequently, the initial value of $D^W(x_i, y_i)$, are zero. Therefore, the only available information at the interior nodes is the one coming from the neighbours. We propagate the neighbouring information by updating the data cost at neighbouring nodes j of the current node $\hat{i}$ (the node with the highest priority) as

$$D^{(t+1)}(x_j, y_j) = D^{(t)}(x_j, y_j) + \min_{x_{\hat{i}}} V(x_{\hat{i}}, x_j), \quad \forall x_j \in \Lambda_j. \tag{5.19}$$

This updated measure can now be used directly in Eq. (5.18) to update $D^W(x_j, y_j)$. Such an update definition is motivated by the update of beliefs within the global framework in p-BP, but we do not require the computation of messages. Note that each node is visited only *once* during label pruning, thus once chosen set of L labels per node remains fixed throughout the rest of the inference algorithm. Therefore, the update is only necessary for *unvisited* neighbouring nodes (with $\nu_j = 0$), because their labels have not been pruned yet (see Algorithm 4).

5.3.2.3 Inference

After labels of each node have been pruned, we can turn to minimizing the energy in Eq. (5.4). We employ here our inference method NCMP, introduced earlier in Section 2.4, to choose one label per node, where the set of labels $\hat{\mathbf{x}}$ over all nodes minimizes the energy in Eq. (5.4). This method uses the message-passing framework, where one joint message, which is a function of beliefs, is sent from the whole neighbourhood to the central node. The message is defined in Eq. (2.34), because this MRF is pairwise, while the belief is defined in Eq. (2.31), where $\phi(x_i, y_i) = \exp\big(-D(x_i, y_i)\big)$. The data cost $D(x_i, y_i)$ and the pairwise potential $V(x_i, x_j)$ are now computed from Eqs. (5.2) and (5.3), respectively, but only for L chosen labels of each node. We initialize the algorithm by first forming the initial mask by maximum likelihood estimation, $\hat{x}_i = \arg\max_{x_i \in \Lambda_i} \phi(x_i, y_i)$, and then we initialize belief of each node by setting it to the value that favours the label of that node in the initial mask. We then

run the algorithm iteratively until the specified number of iterations is reached. Compared to LBP, which is the core of p-BP and which could also be used for inference at this point, NCMP is simpler and faster, and was proved to give good results in other patch-based MRF models (see Chapter 3).

The final inpainting result is formed by assigning each node i its chosen label, which means that we copy the patch of pixel values centred at $\hat{x}_i$ to the positions within the mask centred at i. We can formally write this as

$$g(p+i) = g(p+\hat{x}_i), \quad \forall p \in \Psi, \tag{5.20}$$

where a mask Ψ is a set of positions centred at the origin (see Section 3.3). However, the chosen patches need to be stitched together in the region of overlap. As suggested in [Komodakis 07], we use the minimum error boundary cut [Efros 01] to find the seam along which the transition between two neighbouring patches is the least visible.

5.4 Experiments and results

We evaluate the proposed MBCA method in applications of scratch and text removal, and image editing, i.e., object removal. The reference methods for comparison are chosen from all three categories of image inpainting methods: "greedy", multiple-candidate and global. For all the analysed methods we show the best inpainting result, by optimizing the patch size (where possible). Furthermore, for our method, if not stated otherwise, we use $N_f = 18$ filters (over 3 scales and 6 orientations) and $K = 16$ textons for contextual descriptors, threshold for block similarity $T_b = 0.15$, number of chosen labels $L = 10$ and $N_{iter} = 10$ iterations of the inference algorithm. The threshold for priority T_R is computed as the median value of SSDs computed between each pair of patches in the source region, as suggested in p-BP, just that in our case this source region is constrained and it differs from one block to another. For all the results, we used the division into blocks of adaptive sizes obtained with the proposed top-down splitting procedure. This procedure was conducted until the block size reached $r = 1/4$ of the image size for images in Sec. 5.4.1 and $r = 1/8$ for images in Sec. 5.4.2, because the former images contain a close-up of the object (see Fig. 5.7), thus finer division would not be beneficial.

5.4.1 Experiments and comparisons for scratch and text removal

For the task of scratch and text removal, we use the dataset of four images from [Xu 10] (the top row of Fig. 5.7), where the ground truth is available. The reference methods include the "greedy" approach from [Criminisi 04][1], the commercially available software Content Aware Fill of Adobe PhotoShop, based on [Wexler 07, Barnes 09], the multiple-candidate sparsity-based method

[1] MatLab software from http://www.cc.gatech.edu/∼sooraj/inpainting/.

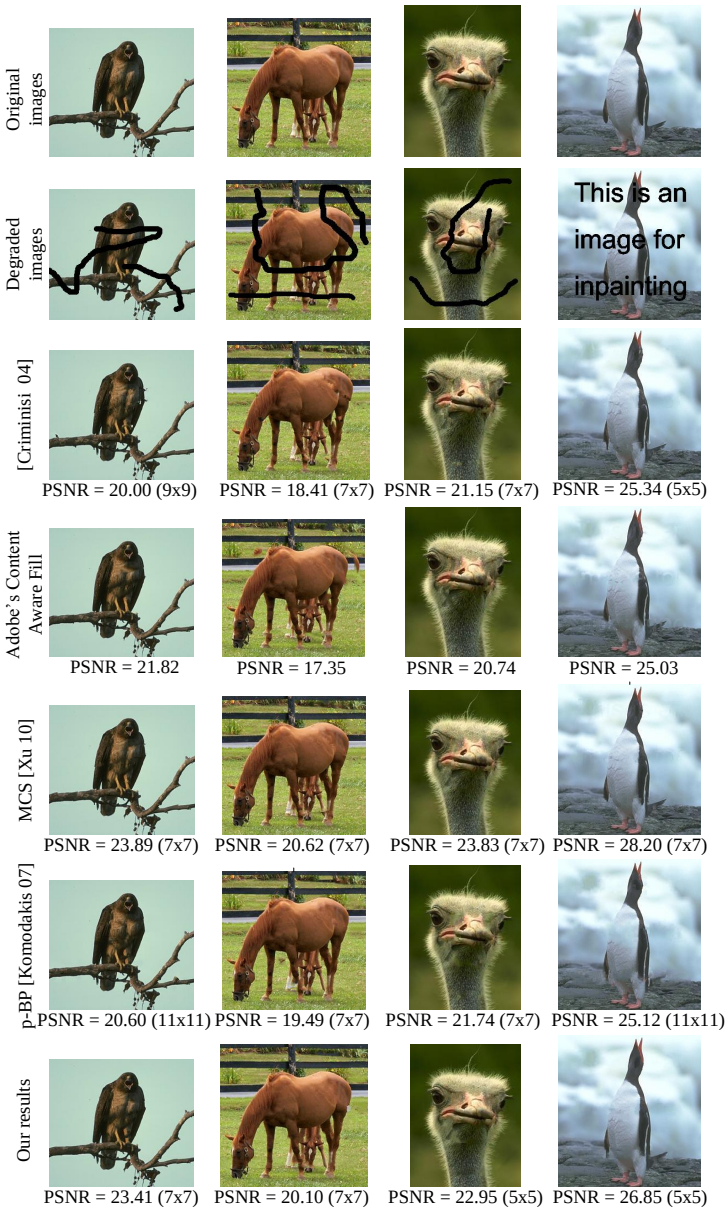


Figure 5.7: Comparison of different inpainting methods for scratch and text removal (see text for details).

(MCS) from [Xu 10][2], and the global method from [Komodakis 07][3], which is most related to ours. Peak signal-to-noise ratio (PSNR) values indicated in Fig. 5.7 are computed only in the target (missing) region as

$$\begin{aligned} \mathrm{PSNR} &= 20 \log_{10} \frac{1}{\sqrt{\mathrm{MSE}}} \\ \mathrm{MSE} &= \frac{\sum_{p \in \Omega} \left(g_{orig}(p) - \hat{g}(p) \right)^2}{\#\Omega}, \end{aligned} \tag{5.21}$$

with the pixel values in the range [0,1], where g_{orig} is the ground truth image and $\hat{g}$ is the resulting inpainted image.[4] The patch size is shown in the parenthesis in Fig. 5.7. We varied the patch size from 5×5 to 13×13 and chose the one with the highest PSNR for each method. Only the Content Aware Fill does not require explicit specification of the patch size.

The results in Fig. 5.7 demonstrate that the proposed MBCA method gives visually pleasing result, with almost no disturbing artefacts. Compared to the methods from [Criminisi 04,Komodakis 07] and Adobe's Content Aware Fill, our method yields the best result for all images, both quantitatively (in terms of PSNR) and qualitatively. Compared to the MRF-based p-BP [Komodakis 07], the increase in PSNR ranges from 0.6 to 2.8dB. Our PSNR values are lower than those of MCS [Xu 10] (with the difference in PSNR ranging from 0.5 to 1.3dB). This can be partly due to the fact that [Xu 10] is ideally suited for this type of problems (thin missing regions), while our method is generally formulated to cope with larger "holes". Nevertheless, this example shows that our method can also deal with scratch/text removal and achieve comparable, and in many cases better results than related and state-of-the-art methods.

5.4.2 Experiments and comparisons for object removal

In this subsection, we deal with a more demanding task of object removal, which requires large missing regions to be inpainted. The bigger the missing region is, the more ambiguity there is on how to fill it in.

In Fig. 5.8, we show the comparison of the proposed MBCA method with the Content Aware Fill and p-BP [Komodakis 07] for the "bungee" image. We can see that our method is more successful in preserving structure in the image. For example, see artefacts on the roof of the building and the grass area below it in the result of the Content Aware Fill (see marked areas), and the border between land and water in the result of p-BP [Komodakis 07] (see marked areas).

[2]Test images and results were received from the authors.

[3]We use our own implementation in MatLab with $L_{min} = 3$, $L_{max=10}$ and 10 iterations of the p-BP algorithm.

[4]We chose to compute PSNR in this way because the same approach was used in [Xu 10], thus we could compare our results with the ones they reported.

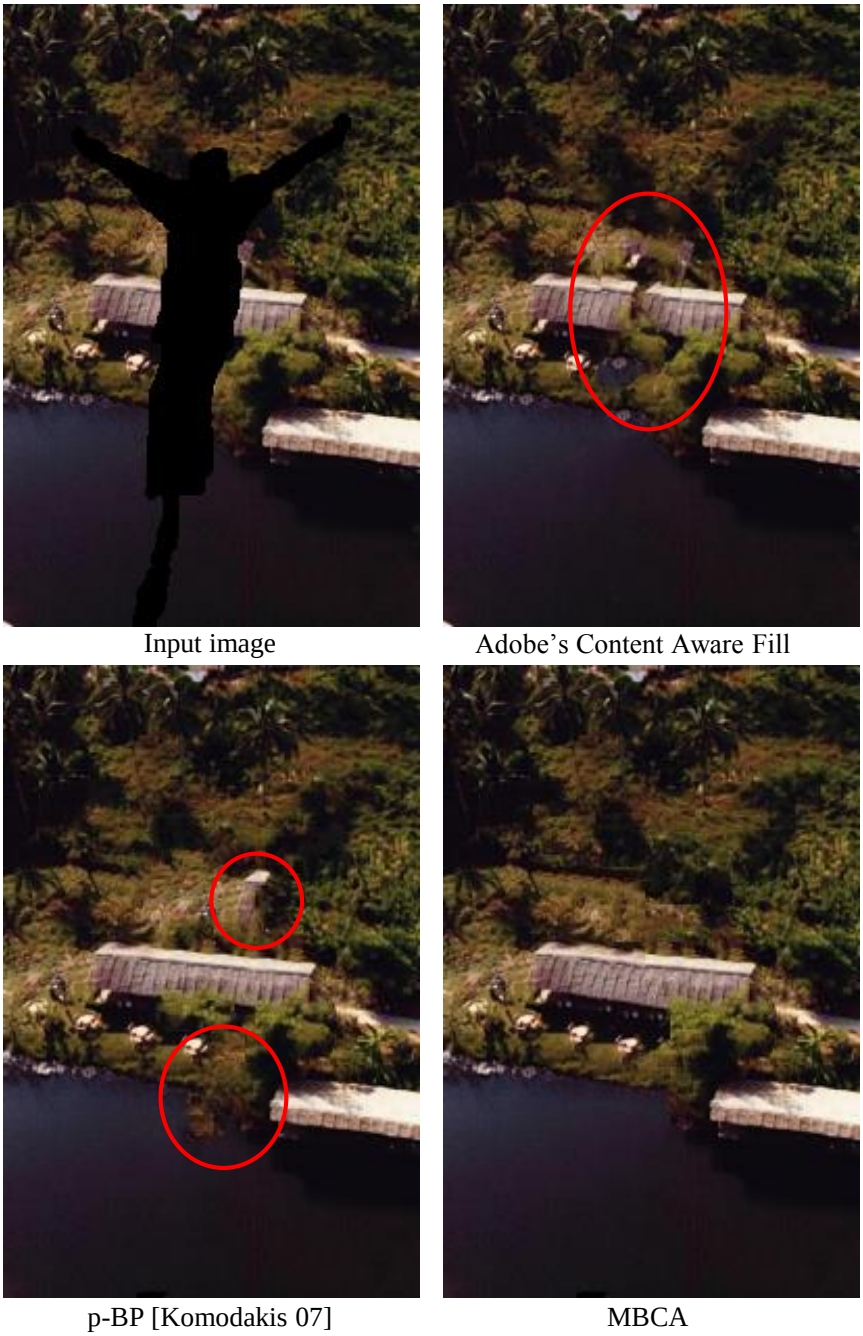


Figure 5.8: Comparison of inpainting results for the "bungee" image. From left to right and top to bottom: input image with the missing region marked in black, result of the Content Aware Fill, result of p-BP [Komodakis 07] (11×11 patches), and result of the proposed MBCA method (13×13 patches).

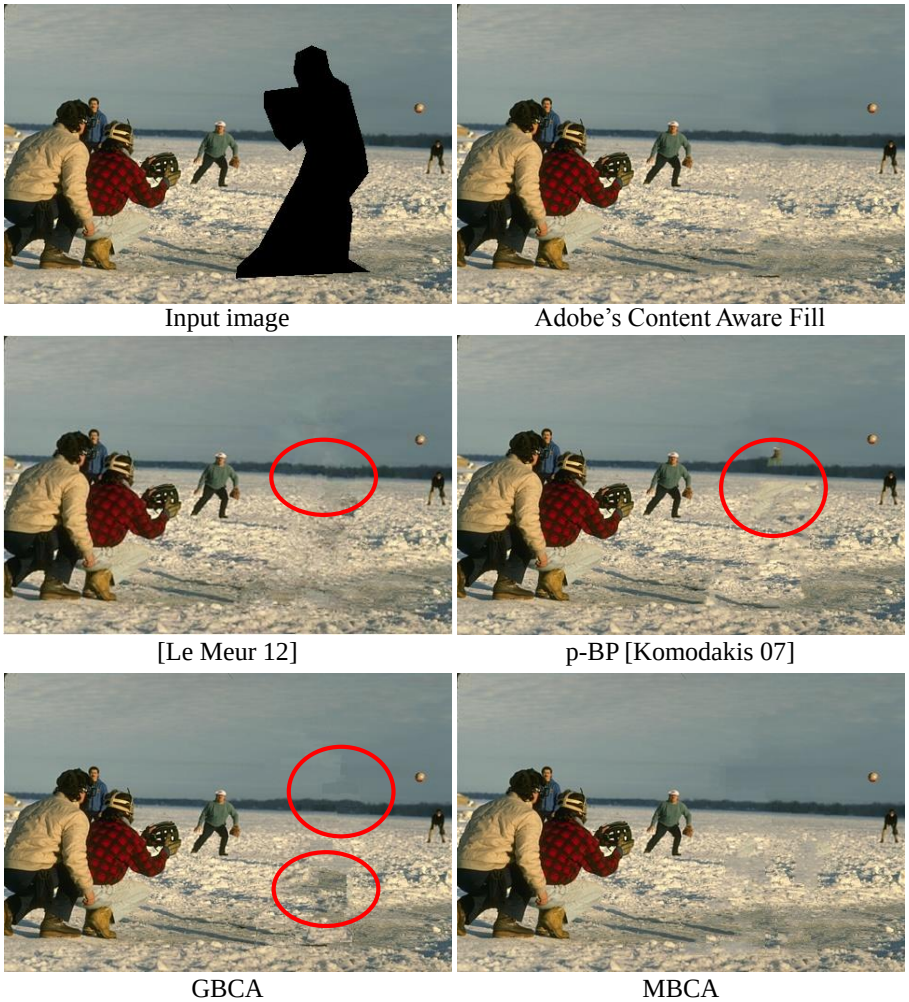


Figure 5.9: Comparison of inpainting results for the "baseball" image. From left to right and top to bottom: input image with the missing region marked in black, result of the Content Aware Fill, result of [Le Meur 12], result of p-BP [Komodakis 07] (7×7 patches), result of our GBCA method (13×13 patches and division into 3×4 blocks of fixed size), and result of the proposed MBCA method (15×15 patches and division into blocks of adaptive sizes from Fig. 5.5b).

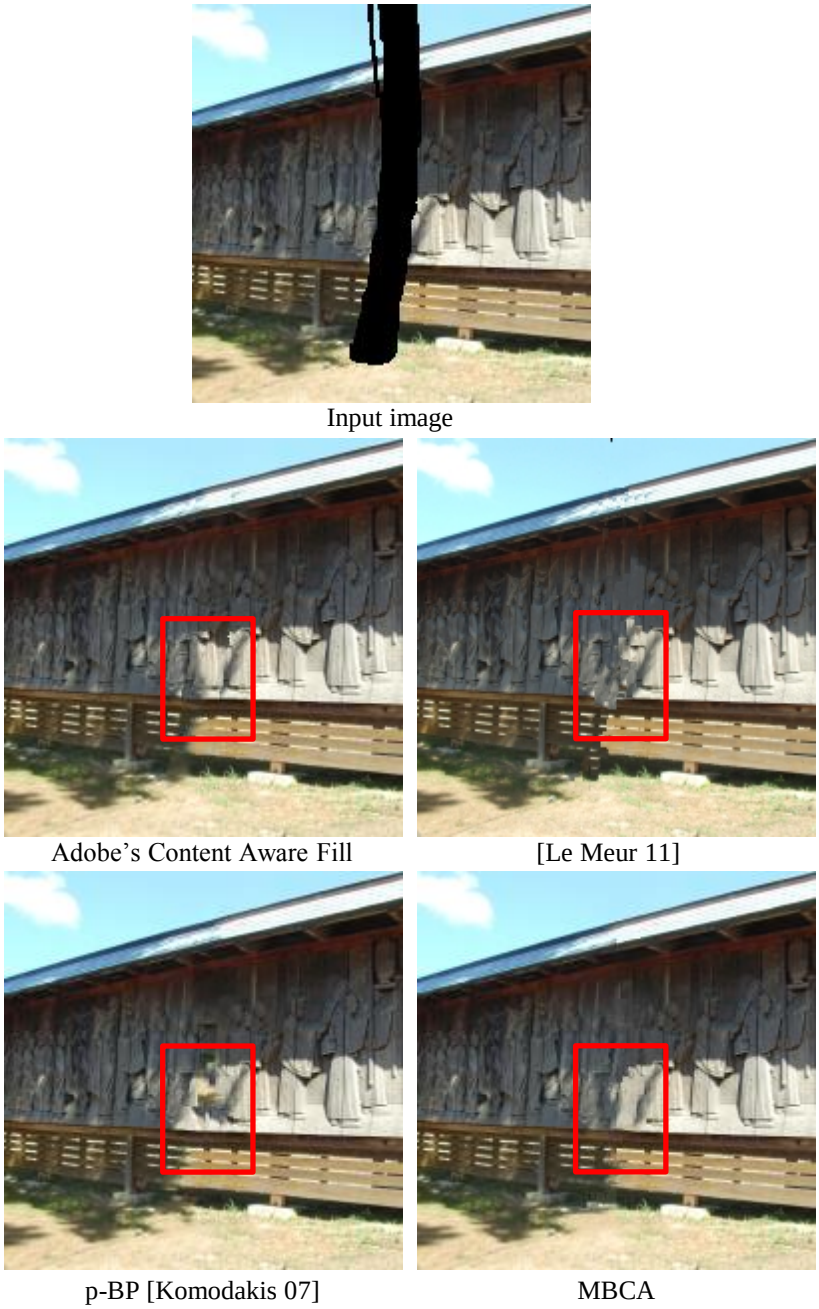


Figure 5.10: Comparison of inpainting results for the "wall" image from [Kawai 08]. From left to right and top to bottom: input image with the missing region marked in black, result of the Content Aware Fill, result of [Le Meur 11], result of p-BP [Komodakis 07] (7×7 patches), and result of the proposed MBCA method (7×7 patches).

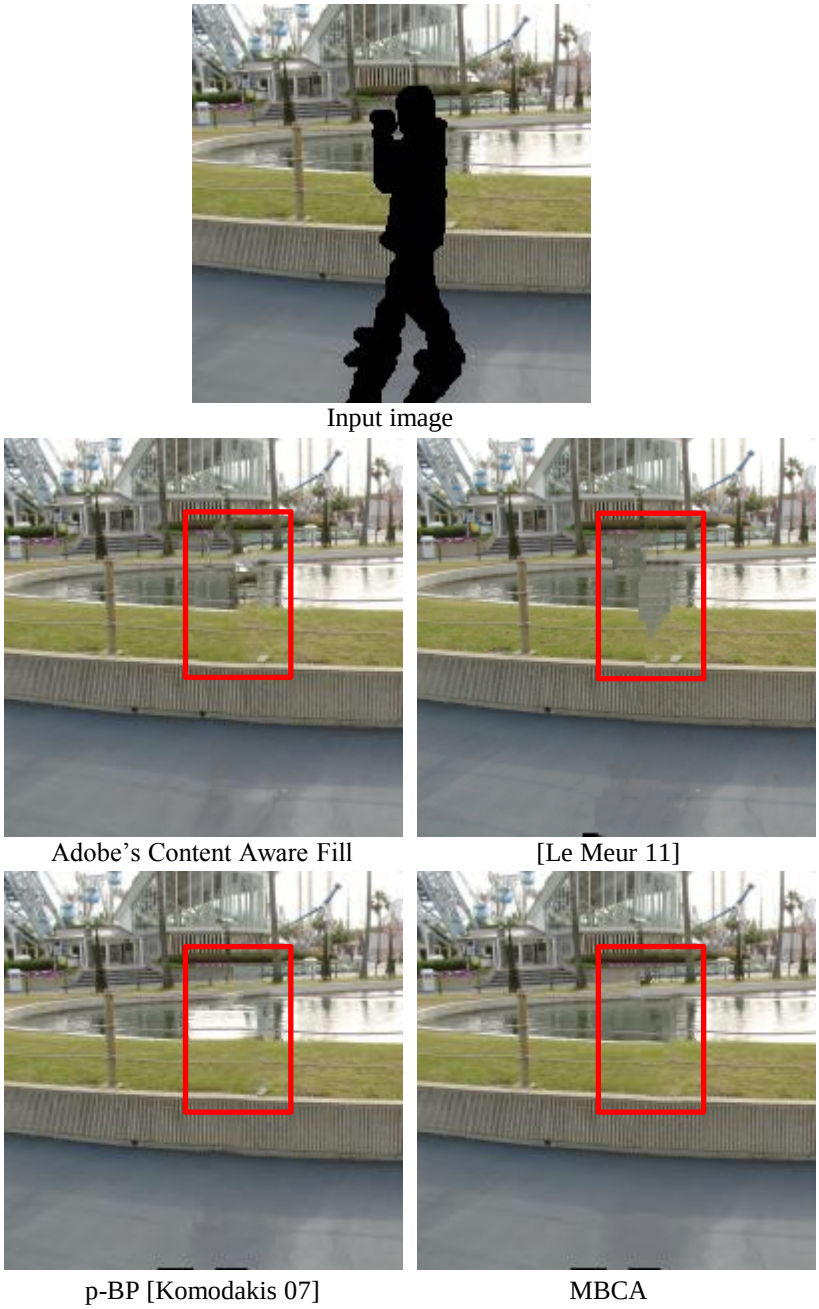


Figure 5.11: Comparison of inpainting results for the "lake" image from [Kawai 08]. From left to right and top to bottom: input image with the missing region marked in black, result of the Content Aware Fill, result of [Le Meur 11], result of p-BP [Komodakis 07] (7×7 patches), and result of the proposed MBCA method (7×7 patches).

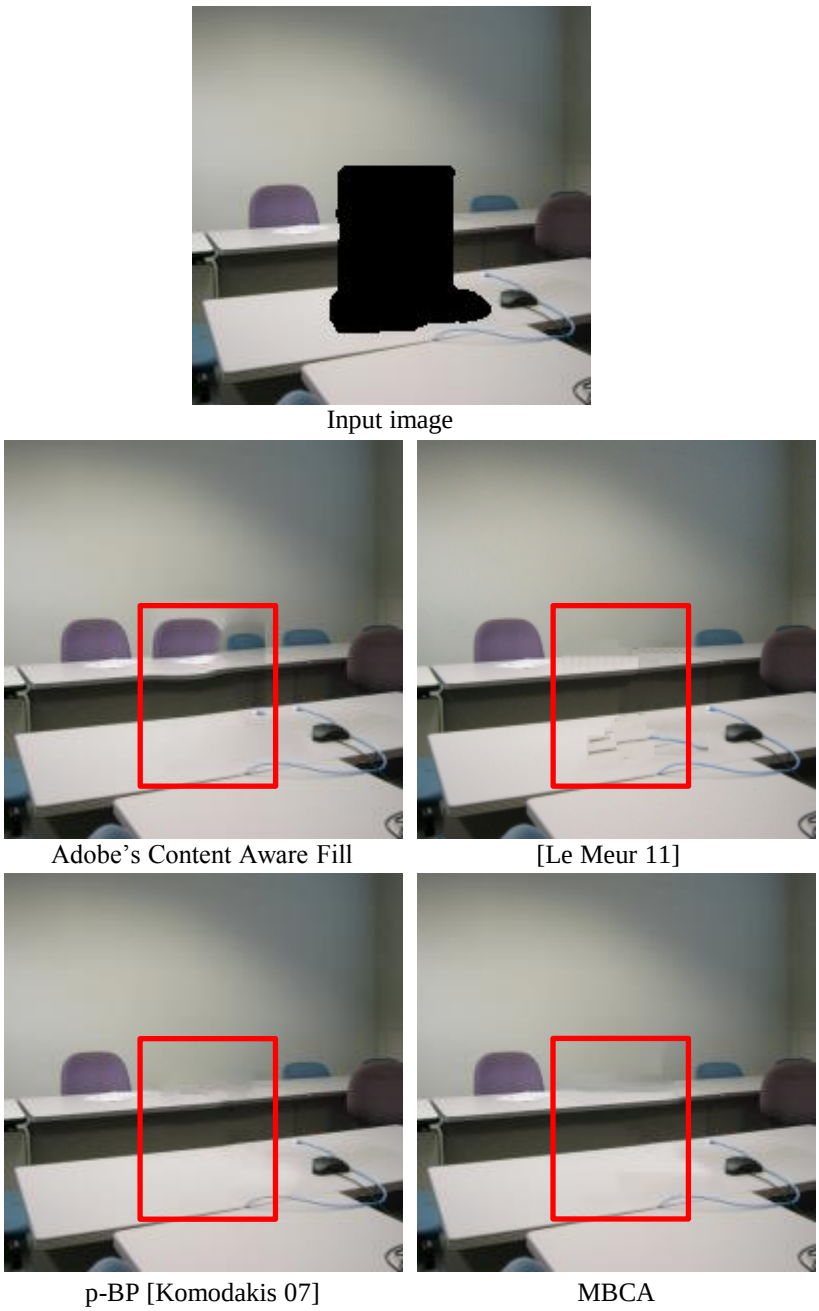


Figure 5.12: Comparison of inpainting results for the "office" image from [Kawai 08]. From left to right and top to bottom: input image with the missing region marked in black, result of the Content Aware Fill, result of [Le Meur 11], result of p-BP [Komodakis 07] (7×7 patches), and result of the proposed MBCA method (5×5 patches).

Table 5.1: Comparison of computation times for different images.

Image (patch size)	p-BP [Komodakis 07]	Proposed method
"bungee" (13×13)	253.88s	113.76s
"baseball" (15×15)	1295.95s	499.5s
"wall" (7×7)	103.33s	18.54s
"lake" (7×7)	138.76s	45.14s
"office" (5×5)	214.23s	58.42s

Table 5.2: Computation times per each phase of the algorithms for the "baseball" image (Fig. 5.9) for 15×15 patches.

Phase	p-BP [Komodakis 07]	Proposed method
threshold computation	144.44s	73.35s
initialization	20.29s	7.88s
label pruning	1126.45s	400.76s
inference	2.67s	0.82s
overhead computations	2.1s	16.69s

The results in Figs. 5.9, 5.10 and 5.11 also show improvements over p-BP [Komodakis 07]. Compared with the Content Aware Fill, our results are better (see Figs. 5.11 and 5.12) or comparable (Figs. 5.9 and 5.10). Finally, we also show the results of two multiple-candidate state-of-the-art methods: the very recent SR-based method from [Le Meur 12] in Fig. 5.9, and the method from [Le Meur 11] in Figs. 5.10, 5.11 and 5.12.[5] The result of our method is comparable to that of [Le Meur 12], although our method preserves better the border between snow and sky. Compared with [Le Meur 11], our method gives superior results on all images in Figs. 5.10, 5.11 and 5.12.

Additionally, Fig. 5.9 shows the advantage of the proposed MBCA method over the GBCA method we proposed in Chapter 4. The texture in the snow contains less artefacts and the border between snow and sky is better preserved. The advantage is also shown in Fig. 5.13 (see marked areas). Besides yielding qualitatively better results, the MBCA method also has the advantage of being more automatic, due to the automatic division into blocks of adaptive sizes and choosing block matches based on the threshold T_b.

Table 5.1 shows the computation times of the p-BP from [Komodakis 07] and the proposed MBCA method, using our own MatLab implementation of both methods on Intel i5-2520M 2.5 GHz CPU with 6GB RAM, for several test images from Figs. 5.8, 5.9, 5.10, 5.11 and 5.12. For fair comparison, we tested the algorithms for the same patch size, which was in this case the one yielding the best result of the proposed method, as indicated in the first column of the table. Note that for the images "wall" and "lake", the best

[5] Results are available on the author's website.

Figure 5.13: Comparison of inpainting results for the "ski resort" image. From left to right: input image with the missing region marked in black, result of our GBCA method (17×17 patches and division into 4×6 blocks of fixed size), and result of the proposed MBCA method (13×13 patches and division into blocks of adaptive sizes).

result of the reference p-BP method was obtained with the same patch size. As we can see from the results, the proposed method is obviously much faster, 2 to 6 times, in all the tested cases, with different image and patch sizes and different sizes of the missing region. However, note that the proposed method is much slower than the earlier proposed GBCA method (see Table 4.1) due to the MRF approach and allowing multiple choice of labels.

Most of the computation time is spent on label pruning (which is the forward pass of the first iteration in p-BP), 87% for p-BP and 80% for our method, as shown in Table 5.2 for the "baseball" image (Fig. 5.9), and this depends on the size of the label set. Therefore, the acceleration of our method is largely due to the use of contextual information, which yields a smaller (constrained) label set, and hence there is less work for pruning. Initialization is also much accelerated due to the same reason. Finally, our inference method is also faster than p-BP (i.e., the backward pass of the first iteration and both passes of subsequent iterations), by about 3-4 times on the "baseball" image, with the same number of iterations ($N_{iter} = 10$) and the same number of pruned labels ($L = L_{max} = 10$). Overhead computations include stitching the patches together and in the case of the proposed method, texton computation, division into blocks of adaptive sizes and block matching. Note also in Table 5.2 that significant amount of time is needed for the computation of thresholds, which include b_{conf} and b_{prune} in p-BP and T_R in our method. However, these threshold computations are for the most images still much faster than label pruning.

5.4.3 Effect of the parameter choice

Let us first comment on the choice of contextual descriptors. Based on extensive experimental evaluation on a number of natural images using the division of the image into fixed-size blocks, we concluded that texton histograms (TH) are a better choice compared with the combination of averaged filter outputs and colour (AFC) from Chapter 4. The main reason is that they allow us to set the threshold T_b for block similarity, i.e., to define as blocks of similar context the ones which yield dissimilarity $\bar{H}^{(l,m)} \leq T_b$, where $\bar{H}^{(l,m)}$ is defined in Eq. (5.9). This threshold is universal, i.e., independent of the block's content and/or size. The value of the threshold depends on the number of filters N_f and the number of textons K, which will be discussed below. An alternative to having a threshold would be to choose for each block a fixed number of block matches as blocks with similar context, as we proposed earlier in Section 4.4.3. However, it is obvious that not all the blocks have equal number of good matches, especially if we look at different images (see Figs. 5.4 and 5.5a). Furthermore, good thresholding performance is crucial for the success of our division into blocks of adaptive sizes (Section 5.2.2).

We illustrate this conclusion in Fig. 5.14. We used $N_f = 18$ filters (across 6 orientations and 3 scales) and $K = 16$ textons and we found $T_b = 0.15$ to be a good choice for a threshold for TH (see blocks within red rectangles in the first row of Figs. 5.14(a) and (b)). We explored also the possibility of setting a threshold T_e for AFC, and we found that for the above mentioned filtering parameters $T_e = 2 \times 10^{-4}$. We can see in Fig. 5.14 that AFC is less robust to the block's content, which results in a selection of many wrong block matches (see blocks within red rectangles in the second row of Figs. 5.14(a) and (b)). This behaviour is less desirable than having a smaller number of matches, which would all be correct, as in the first row of Figs. 5.14 (a) and (b). Furthermore, we can see in Fig. 5.14 that TH produces better block-matching results in this example than AFC, in the sense that more similar blocks are found and they are better ordered.

One could argue that the threshold T_e for AFC could be set to some lower value to achieve similar block matching as with TH. However, in the second row of Fig. 5.15, we can see that $T_e = 2 \times 10^{-4}$ is already too low because it chooses only the current block as the constrained source region, which can be too small to find well-matching patches. On the other hand, $T_b = 0.15$ chooses a sufficient number of good matches also for this image. Therefore, we can conclude that indeed TH is more robust to block's content, especially across different images.

We also made experiments for TH with $N_f = 24$ filters (over 3 scales and 8 orientations) and $K = 32$ textons, which requires the threshold T_b to be adapted to 0.2 to achieve similar block-matching result. The comparison of divisions into blocks of adaptive sizes is shown in the top row of Fig. 5.16. We can see that using more filters gives finer division, which is the case for most images. This is sometimes justified by the content of the image, but most often this division is too fine and leaves more unreliable blocks (see Eq. (4.7)),

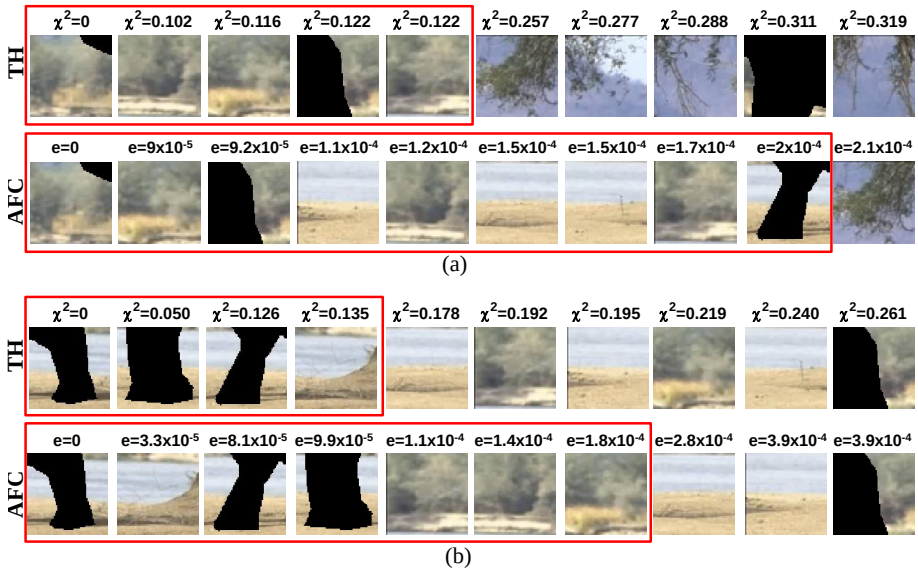


Figure 5.14: Examples of block matching for two blocks from the "elephant" image (Fig. 5.4). (a) and (b): current block (in the first column) and the corresponding 10 matches (including the current block itself) obtained by comparing different features: TH and AFC. Block matches are ordered from left to right from most to least similar. Above each block match, the dissimilarity value is shown (χ^2 for TH and e as SSD of feature vectors for AFC). The red rectangles mark the blocks whose dissimilarity is lower than the threshold: $T_b = 0.15$ for TH and $T_e = 2 \times 10^{-4}$ for AFC.

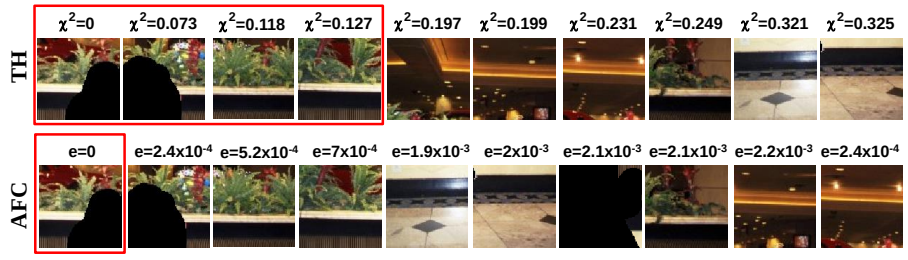


Figure 5.15: Example of block matching for one block from Fig. 4.8. The current block (in the first column) and the corresponding 10 matches (including the current block itself) obtained by comparing different features: TH and AFC. Block matches are ordered from left to right from most to least similar. Above each block match, the dissimilarity value is shown (χ^2 for TH and e as SSD of feature vectors for AFC). The red rectangles mark the blocks whose dissimilarity is lower than the threshold: $T_b = 0.15$ for TH and $T_e = 2 \times 10^{-4}$ for AFC.

which is often undesirable. On the other hand, as we can see from the results in the bottom row of Fig. 5.16, similar results can be achieved with both sets of

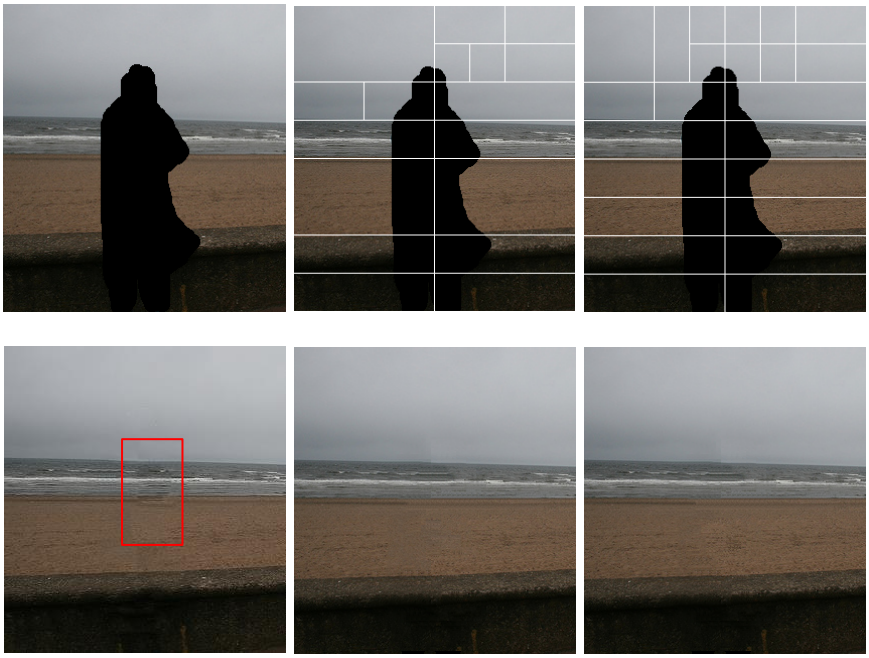


Figure 5.16: Effect of filtering parameters on the division into blocks of adaptive sizes and the final inpainting result. Top row (from left to right): input image with the missing region marked in black, division obtained with $N_f = 18$, $K = 16$ and $T_b = 0.15$, and division obtained with $N_f = 24$, $K = 32$ and $T_b = 0.2$. Bottom row (from left to right): result of the MCS method [Xu 10], result of the proposed MBCA method using the division above it and 11×11 patches, and result of the proposed MBCA method using the division above it and 11×11 patches.

parameters, which are better than the result of the MCS method from [Xu 10] (see the marked area). In the end, we found the first set of parameters, i.e., $N_f = 18$, $K = 16$ and $T_b = 0.15$, to be a good and stable choice for different images.

Finally, we comment on the choice of L, i.e., the number of labels kept after label pruning. We found $L = 10$ to be a good trade-off between algorithm's computational complexity and quality of the result, because more chosen labels means longer computation time. In the method from [Komodakis 07], L_{max} was allowed to go up to 50, but we consider this number to be unnecessarily high in our approach, considering that our label set is already limited due to the context-aware label selection.

5.5 Conclusion

In this chapter, we extended our work on context-aware patch-based inpainting. We introduced several contributions compared to Chapter 4. First of all, we introduced a novel context representation within blocks of adaptive sizes using contextual descriptors in the form of normalized texton histograms. Normalized texton histograms have been widely used for image segmentation and texture classification, but to our knowledge, they have never been used for image inpainting before. We showed the advantages of these contextual descriptors compared to averaged filter outputs and colour used in Chapter 4. Additionally, to divide the image into blocks of adaptive sizes, a novel top-down splitting procedure was introduced, which is also based on contextual descriptors.

We applied this improved context-aware approach within a novel MRF-based inpainting method (MBCA method) in order to reduce the number of possible labels per MRF node and choose them in such a way that they better fit the surrounding context. MRF-based, i.e., global approach, overcomes some of the limitations of the "greedy" approach, which we used in Chapter 4. We also proposed a simple and efficient way to perform optimization by first pruning the labels, and then separately employing the inference method to obtain the final inpainting result. Labels are pruned for each node separately by visiting the nodes in the order of priority, which is necessary to ensure that the node has sufficient information based on which it can choose its labels. Label pruning in the order of priority was motivated by the original MRF-based approach for inpainting, which we also described in detail in this chapter. However, our approach has several major differences, as we pointed out earlier in the chapter.

We evaluated the proposed method on two example applications: scratch and text removal and photo-editing. Results demonstrated the benefits of our approach in comparison with state-of-the-art methods in terms of quality and additionally, in comparison with a related MRF-based method, in terms of speed. We also qualitatively compared the results of the proposed MBCA method with the earlier proposed GBCA method from Chapter 4, and we showed that they are more visually pleasing, at the expense of the higher computation time.

In both this chapter and Chapter 4, we evaluated our inpainting methods on natural images from Berkeley segmentation database[6], which is often used as a data set in image processing and analysis. Our method could also be used for error concealment in video and multi-view synthesis, e.g., for free viewpoint television (FTV). Rather than applying the method "as is", e.g., frame-per-frame, we could gain more by exploiting temporal and/or depth information. Some of these possibilities are discussed in Section 7.2. Finally, in the next chapter, we consider another special application of image inpainting for crack removal in digitized paintings.

[6] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench

This work resulted in one journal paper submission [Ružić 13b], while some parts were presented in two conference publications [Ružić 12b, Ružić 13a].

6

Crack removal in artwork

Today, museums and galleries are digitizing their artwork for the purpose of archiving and dissemination. Sometimes, these digitized paintings are available online in high resolution, thus enabling a large audience to appreciate the paintings and their finest details. Furthermore, this digitization has opened the door for a new application of digital image processing and analysis focused on art investigation.

In this chapter, we are focusing on virtual restoration of artwork, which enables the removal of signs of ageing of a painting, such as cracks. Specifically, we address the problem of crack removal in the digitized versions of the *Adoration of the Mystic Lamb*, also known as the *Ghent Altarpiece*. Crack removal involves filling the detected cracks with appropriate content. Hence, it can be regarded as a special application of image inpainting, which we studied earlier in Chapters 4 and 5. However, cracks in old paintings are particular in a number of ways, which makes crack inpainting a challenging problem, where most "off-the-shelf" general inpainting methods fail. Therefore, we develop a novel crack inpainting method (Section 6.6), which incorporates our earlier ideas of context-aware inpainting, but also specifics of the application.

6.1 Introduction

Image processing for art investigation can be roughly divided into two large groups of methods. One group of digital image analysis methods focuses on characterizing the style of the painter [Platiša 11] to, e.g., facilitate artist identification and forgery detection [Johnson 08,van der Maaten 10,Daubechies 12]. The other group of methods focuses on virtual restoration of digitized paintings with the goal of improving the visual experience or facilitating art historical and iconographical analysis. Virtual restoration usually aims at removing the signs of ageing in paintings, such as stains, artefacts and cracks. In this work, we will focus on crack removal.

6.1.1 Cracks in old paintings

Breaking of the paint layer, called *craquelure* or *cracks*, is one of the most common deteriorations in old paintings (see examples in Figs. 6.2, 6.4 and 6.3). It is a sign of the inevitable ageing of materials, and it constitutes a record of their degradation. The formation and the extent of cracks is caused and influenced by different factors, by which they can be divided into age, mechanical, premature and varnish cracks.

Age cracks are mainly caused by climate changes, such as variations in temperature, relative humidity or pressurization (e.g., during transport via air when the pressure in the cargo compartment is lower than in the cabin) [Abas 03]. Mechanical cracks result from external impacts, such as vibrations during transport and human handling. Age-related and mechanical cracks can affect the entire paint layer structure, including both the preparation layer and the paint layers on it. On the other hand, premature cracks [Mohen 06] are more dull-edged than those formed by ageing, and originate in only one of the paint layers. They generally reveal a defective technical execution at the painting stage, such as not leaving enough time for a layer to dry, or applying a layer that dries faster than the underlying one. Finally, varnish cracks are formed in the varnish layer, which protects the paint layer and is originally transparent. During ageing, this varnish layer loses its transparency and becomes greenish or yellowish. Since also the paint layer ages, varnish is no longer capable of keeping the paint layer intact, thus cracks begin to form [Abas 04].

Crack patterns can be of different shapes: rectangular, circular, web-shaped, unidirectional, tree-shaped or even completely random [Cornelis 13]. The appearance of cracks and the whole crack pattern depends on the choice of materials and methods used by the artist. This makes cracks useful for judging authenticity, as is proposed in [Bucklow 97]. Cracks can also assist conservators by providing clues on the causes of degradation of the paint surface. This can be used for degradation monitoring of the paint layer, or a more in-depth study on factors that contribute to the formation of cracks, so that steps can be taken to reduce them [Abas 04]. The potential of using cracks as a *non-invasive* means of identifying the structural components of paintings is highlighted in [Bucklow 98]. The correlation between the network of cracks on the surface and the structure of the panel below is also investigated in [Mohen 06] by using multi-layered X-ray radiography. An area which is thought to be of great interest to art conservation is content-based analysis, where cracks are used for content-based retrieval of information from image databases [Abas 03].

In this chapter, we consider cracks to be an undesired pattern in digitized paintings, which we would like to remove by the means of *virtual* restoration. This virtual removal employs digital image processing techniques and it consists of two steps: 1) detection of cracks, and 2) "filling" the detected cracks such that they are no longer visible. Although cracks are inherent to our appreciation of these paintings as old and valuable, they deteriorate perceived image quality. Cracks become especially prominent and disturbing when zooming in on details of the high-resolution (HR) scans of the paintings [Pižurica 13],

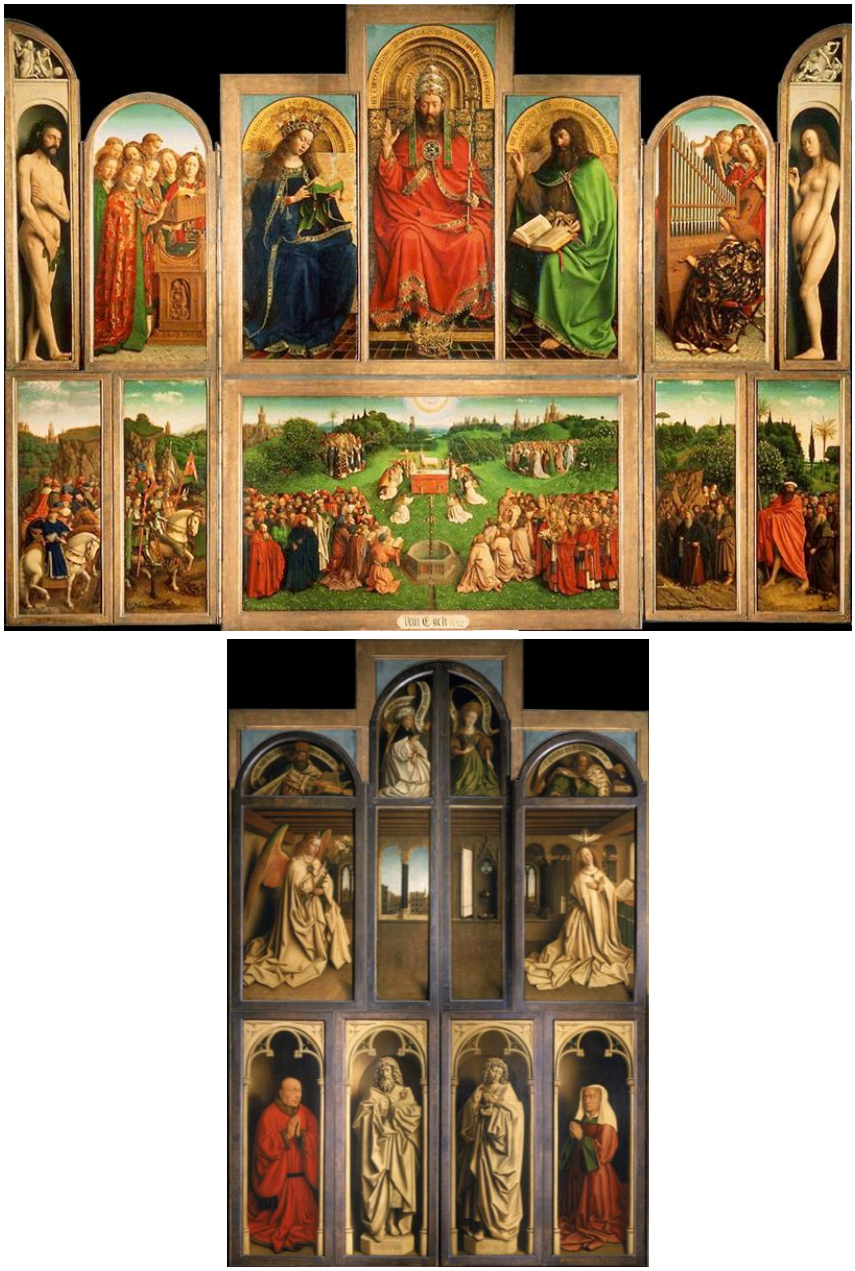


Figure 6.1: The opened (top) and the closed view (bottom) of the Ghent Altarpiece.

which are nowadays made available for the larger audience through websites

such as the Google Art Project[1] or the Closer to Van Eyck[2]. Since cracks are never removed in the actual, physical restoration of the painting, their virtual removal is the only way to obtain the painting as it used to appear before the ageing process. This does not only make the appearance more pleasing, but it can also be of interest in psychovisual studies, to analyse how our perception of the painting is affected when observing it in its initial state. Moreover, crack removal can facilitate art historical and iconographical analysis of paintings.

6.1.2 Case study: the Ghent Altarpiece

In this work, we focus on the difficult problem of crack inpainting in the *Adoration of the Mystic Lamb*, also known as the *Ghent Altarpiece* (see Fig. 6.1). The polyptych consisting of 12 panels, dated by inscription 1432, was painted by Jan and Hubert van Eyck, and is considered as one of the most important masterpieces in the world. It is still located in the Saint Bavo Cathedral in Ghent, its original destination.

We got involved in this research on the initiative of prof. Ingrid Daubechies from the Mathematics Department, Duke University, USA, who brought us into contact with prof. Mark de Mey from the Royal Flemish Academy of Belgium (KVAB)[3], Belgium, and prof. Maximiliaan Martens and Emile Gezels from the Department of Art, Music and Theatre Sciences, Ghent University, Belgium. Together with prof. Ann Dooms and ir. Bruno Cornelis from the Vrije Universiteit Brussel, Belgium and, ir. Ljiljana Platiša from the IPI group, we started collaboration on this project, aimed at developing image processing tools for art investigation.

In the Ghent Altarpiece, as in most 15^{th} century Flemish paintings on Baltic oak, fluctuations in relative humidity cause the wooden support to shrink or expand, thus forming age cracks. These cracks are particular in a number of ways:

1. The width and length of cracks ranges from very narrow, barely visible hairline structures to larger areas of missing paint (see enlarged detail on the far right of Fig. 6.2).

2. Depending on the painting's content, cracks appear as dark thin lines on a bright background or vice versa, bright thin lines on a darker background (see the enlarged detail in the bottom right of Fig. 6.4).

3. Often, cracks have very similar characteristics, e.g., colour and width, as brush strokes that depict fine details, thus it is difficult to make a distinction between them in some parts of the image (see the eye lashes in Adam's eye in the bottom left of Fig. 6.4, and the letters of the book in Fig. 6.3).

[1] http://www.google.com/culturalinstitute/project/art-project

[2] http://closertovaneyck.kikirpa.be/

[3] At the time of this research, prof. Mark de Mey was affiliated with the Flemish Academic Centre for Science and the Arts (VLAC), Belgium.

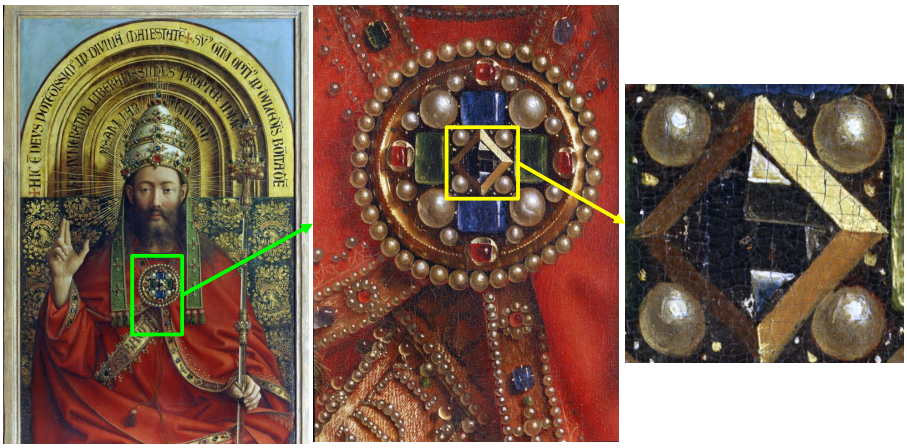


Figure 6.2: The *God the Father* panel and the enlarged image of the locking pin.

Figure 6.3: The *Annunciation to Mary* panel and the enlarged image of the book. The height of one letter on the painting is approximately 0.17cm.

4. Some of the cracks are surrounded with bright borders (visible in the enlarged details in Figs. 6.2 and 6.4, and mostly visible in the enlarged detail on the far right of Fig. 6.3), which cause incorrect and visually disturbing inpainting results. These borders are caused by either the reflection of light on the inclination of the paint caused by the crack on the varnish layer, or the exposure of the underlying white preparation layer due to the accidental removal of the surface paint due to wear or after cleaning.

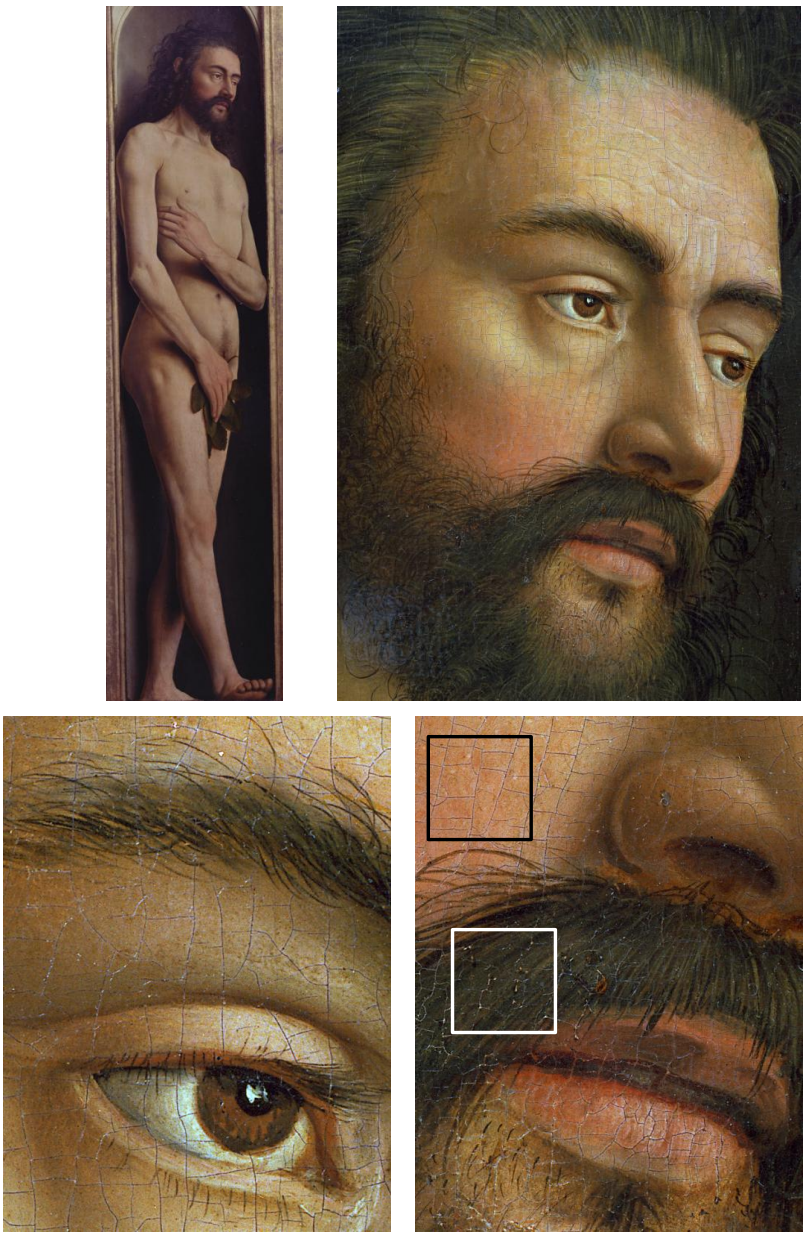


Figure 6.4: The *Adam* panel and the enlarged image of Adam's head, eye and mouth. The width of the mouth on the painting is approximately 2.86 cm. Black and white squares on the detail of Adam's mouth illustrate different types of cracks: dark cracks on a bright background (black square) and bright cracks on a dark background (white square).

As we mentioned earlier, crack removal does not only enhance the visual experience and aids psychovisual studies, but it can also facilitate art historical analysis. In the Ghent Altarpiece, for example, of great interest to art historians is the text in the book in the *Annunciation to Mary* panel (Fig. 6.3), because its paleographical deciphering can give new clues to the iconographical and theological meaning of this artwork. Since this panel is heavily damaged by cracks, which, in addition, have similar characteristics as the letters, crack removal can improve the legibility of the text and thus, aid paleographers in its deciphering. Finally, it is important to mention that currently the physical restoration of the Ghent Altarpiece is taking place, and it will last for (at least) another four years. However, in this restoration, the cracks will not be physically removed, thus our virtual restoration is a valuable "complement" to the actual ongoing physical restoration.

In this work, we used HR scans of the original photographic negatives (Kodak Safety Film 13×18 cm) taken by a professional photographer, the late Rev. Alfons Dierick. This material is currently preserved in the Alfons Dierick-fonds archive of Ghent University. The images were acquired under different conditions, i.e., different field of view, lighting circumstances and chemicals used to develop the negatives. Moreover, the negatives were scanned at different resolutions and with different scanning hardware. As a consequence, the quality of the images varies significantly making a general, automatic crack detection and inpainting a very difficult problem. Now, there is also a Closer to Van Eyck[4] archive, which consists of images of the painting acquired in four different modes: digital macro photography, digital macro infra-red photography, infra-red reflectography and X-radiography. However, this archive was not at our disposal during our research. We intend on working on this new database in our future work.

6.2 Related work

Automatic detection of crack-like patterns or similar elongated structures is well studied in many applications of digital image processing. Examples, other than old paintings, include medical images of veins and vessels [Zana 01], images of fingerprints, and satellite imagery of rivers and roads. Some common principles, which are often referred to as *ridge-valley structure extraction* [López 99], can be used to extract or detect these crack-like patterns in order to separate them from the rest of the image. An overview of different crack detection techniques can be found in [Abas 04]. These include different types of thresholding, the use of multi-oriented filters (e.g., Gabor filters, see Section A.2) and a variety of morphological transforms.

The second step of virtual restoration of digitized paintings is to fill in the cracks in a visually plausible way. For this purpose, image inpainting can be used (see Chapters 4 and 5), where the detected cracks are treated as missing

[4] http://closertovaneyck.kikirpa.be/

regions that need to be filled in. In Chapter 4, we made a distinction between two large groups of methods: geometry-based and patch-based methods. While patch-based methods (in most cases) fill in the missing region patch-by-patch, geometry-based methods aim at replacing one missing pixel at the time, thus they can be regarded as pixel-based.

Crack detection and removal is related to the detection and removal of scratches and other artefacts from films [Joyeux 99,Kokaram 95a,Kokaram 95c], but these methods exploit information from several frames and thus cannot be applied directly to the restoration of old paintings. There are several methods in literature that explicitly address the complete virtual restoration problem in old paintings [Giakoumis 98, Barni 00, Giakoumis 06, Solanki 09, Spagnolo 10]. The method from [Barni 00] is based on a semi-automatic crack detection procedure, where users need to select a crack point. The algorithm will then track other suspected crack points based on two main features, namely absolute grey level and crack uniformity, under the assumption that the cracks are characterized by a uniform grey colour, which is darker than the background. Once the algorithm has completely detected cracks, they can be removed by interpolation. The authors in [Barni 00] propose to use interpolation based on weighted averaging of the *known* pixels within some radius from the crack point [Franke 82].

In [Giakoumis 98, Giakoumis 06, Solanki 09, Spagnolo 10], crack patterns are detected by thresholding the output of the morphological top-hat transform [Meyer 79]. Cracks are subsequently separated from brush strokes by: 1) using the hue and saturation information in the HSV or HSI colour space and feeding it to a neural network, or 2) letting a user manually select seed points. Finally, the cracks are inpainted using order statistics filtering for interpolation [Giakoumis 06, Solanki 09] or controlled anisotropic diffusion [Giakoumis 06]. Order statistics filters proposed for this purpose are the median and the mean filters and their variations. Controlled anisotropic diffusion represents a modification of anisotropic diffusion [Perona 90], which takes into account crack orientation. This means that the diffusion is applied only in the direction perpendicular to the crack direction. Both order statistics filtering and controlled anisotropic diffusion can be considered as very simple pixel-based inpainting methods, from which controlled anisotropic diffusion produces better crack inpainting results, as shown in [Giakoumis 06]. Finally, in [Spagnolo 10] patch-based texture synthesis is applied to fill in the cracks in combination with median filtering, in the sense that if a well-matching replacement patch cannot be found, the central pixel of the patch is replaced by the median value of the pixels in the patch.

Another virtual restoration method was presented in [Hanbury 03], which focuses on detecting and removing cracks from infra-red reflectograms. These types of images show the underdrawing, i.e., the basic concept of the painting that is drawn by the artist on the ground layer. The authors assume that the cracks are thinner than the brush strokes and that they have a favoured orientation. Then they use viscous morphological reconstruction [Serra 99] to detect and fill in the cracks in one step.

Related work on virtual restoration of old paintings is also presented in [Pei 04, Papandreou 08], where the goal is to remove undesirable patterns, not specifically cracks, which are manually indicated. For example, the method in [Pei 04] removes stains and artefacts in Chinese paintings, which are caused by ageing, by the means of texture synthesis similar to [Efros 99]. To preserve linear structures in the paintings, it is required to *manually* continue the structures through the missing (damaged) region. On the other hand, Papandreou et al. [Papandreou 08] propose an inpainting method based on the hidden Markov tree model in the complex wavelet domain [Kingsbury 01]. They apply it on the wall paintings to fill in the gaps, which arise from making a mosaic of small fragments of the painting.

6.3 Crack detection

A hybrid approach for crack detection in the Ghent Altarpiece was proposed in [Cornelis 13], which deals with specific problems of cracks in this painting described earlier in Section 6.1.2. This approach employs three different crack detection techniques: filtering with oriented elongated filters [Poli 97], a multi-scale morphological top-hat transform [Meyer 79], and image reconstruction from learned dictionary representations using K-SVD [Aharon 06]. Each of these techniques has its own strengths and weaknesses:

- Oriented elongated filters detect most of the cracks of various widths, but they also detect other elongated structures, such as brush strokes and image objects.
- A multi-scale morphological top-hat transform reduces the number of falsely detected cracks and it makes a distinction between fine and coarser cracks, thus improving on the classical top-hat transform. However, some of the very fine cracks can remain undetected due to the way the crack maps are combined over the scales.
- The K-SVD approach yields a smooth crack map, but the results depend on the ability of the learned dictionary to represent cracks of different widths and orientations.

Fig. 6.5 shows an overview of the hybrid crack detection approach (for the details of each of the three techniques, see [Cornelis 13]). The method also includes pre- and post-processing. Pre-processing is necessary to enhance the detection performance in low-contrast areas, thus it comprises of a local contrast enhancement step. Post-processing, on the other hand, differentiates cracks and brush strokes falsely detected as cracks by using a semi-automatic K-means clustering-based procedure [Duda 73]. The idea is to divide a crack map into smaller segments (see [Cornelis 13] for details), which are clustered based on the combination of features like colour, physical properties (length, orientation and eccentricity), colour of the surrounding region and spatial density. From the resulting clusters, the ones that correspond to falsely detected

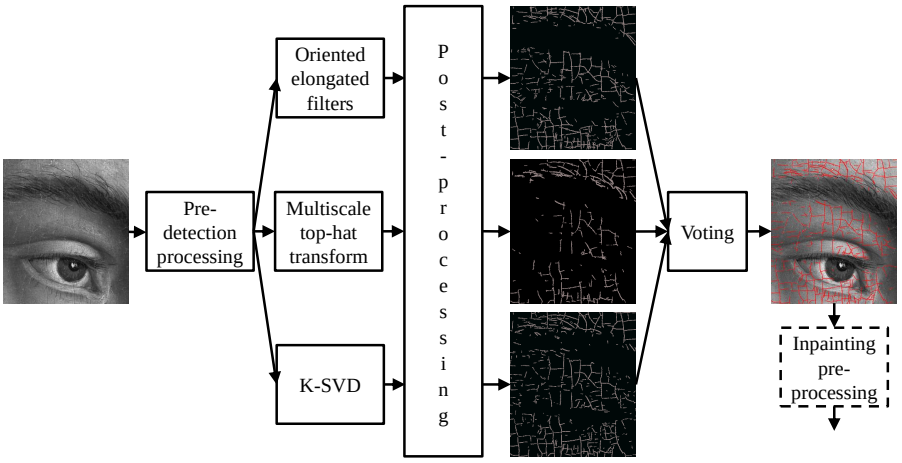


Figure 6.5: Overview of the crack detection procedure [Cornelis 13]. Inpainting pre-processing (within dashed block) is optional, applied only when bright borders around cracks are very prominent (e.g., in the image from Fig. 6.3).

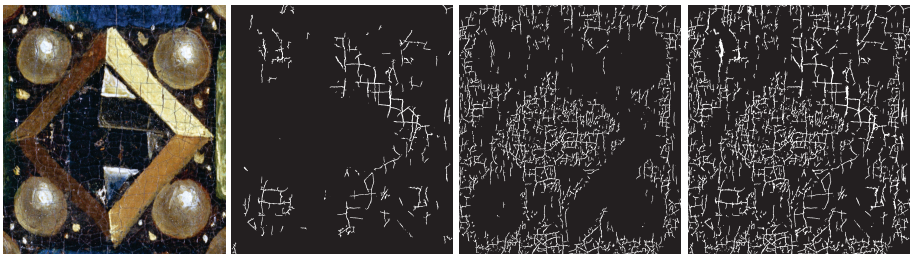


Figure 6.6: Crack detection results [Cornelis 13]: (a) original, (b) dark crack map, (c) bright crack map, and (d) combined dark and bright crack maps.

brush strokes are manually removed. These small segments are also used in the final step of the algorithm: a specially designed voting procedure, which combines the post-processed crack maps obtained by the three techniques into one single crack map. This voting basically marks the segment as a crack if most of its pixels are detected as cracks with at least two detection methods.

The above described method can be applied to detect both dark cracks on a light background, the result being a *dark crack map*, and bright cracks on a darker background, resulting in a *bright crack map*. To obtain the final crack map, dark and bright crack maps are simply combined (see an example in Fig. 6.6).

Recall that in Section 6.1.2, we outlined the existence of whitish/bright borders around the cracks as one of the problems of the cracks in the Ghent Altarpiece, which can negatively influence inpainting result, as

Figure 6.7: Detecting white borders [Cornelis 13]. Left: original colour image overlapped with a square, which shows the values in the blue plane of the RGB image representation. Right: detection result in the blue plane.

we will show later in Section 6.5. These borders in most of the images can be detected with the bright crack map. However, in some cases, e.g., in the book in Fig. 6.3, the bright crack map is insufficient because the borders are much wider than the cracks. To address this problem, an optional step called *inpainting pre-processing* is introduced, which extends the crack map with corresponding bright regions by using their high response in the blue plane of the RGB representation of the image, because it was experimentally determined that in this plane they are the most prominent, thus the easiest to detect (see Fig. 6.7).

6.4 Patch-based methods in crack inpainting

In the process of virtual restoration of digitized paintings, cracks, once detected, can be treated as missing regions that need to be filled in. Therefore, removal of cracks falls into the category of image inpainting (see Chapter 4 for an overview of inpainting methods). *Crack inpainting* methods considered in literature so far are mostly pixel-based, and include order statistics filtering [Giakoumis 06, Solanki 09], controlled anisotropic diffusion [Giakoumis 06] and interpolation [Barni 00]. In [Spagnolo 10], a patch-based texture synthesis method was used (see Section 6.2 for a more detailed overview of related methods).

In Chapters 4 and 5, we proposed two context-aware patch-based inpainting methods: greedy block-based context-aware (GBCA) and MRF block-based context-aware (MBCA) method, respectively. We shall compare these approaches to the best-performing crack inpainting method among the afore-

mentioned ones from literature. In particular, we use as a reference controlled anisotropic diffusion (CAD), which was reported in [Giakoumis 06] to outperform other pixel-based crack inpainting methods, including order statistics filtering. Additionally, we shall test the GBCA method with two different priorities: 1) our orientation-based priority (GBCA-O) defined in Eq. (4.13), and 2) the confidence-based priority (GBCA-C). The confidence-based priority represents the confidence term from the method in [Criminisi 04] (see also Section 4.3.3). It is computed based only on the relative number of existing pixels within the target patch:

$$R(p) = \frac{\sum_{q \in \Psi} D_c(q+p)}{\#\Psi}, \tag{6.1}$$

(using the previously introduced notations for patch-based methods in Chapters 3 and 4). The confidence $D_c(p)$ is initially set to zero for the pixels in the target region Ω and to one for the pixels in the source region Φ. After the missing pixels in the target patch are filled in (see step 8 in Algorithm 2), the confidence values at those pixels are updated by the value of priority of that target patch, computed as in Eq. (6.1). Therefore, this confidence-based priority does not consider the presence of image structures. The orientation-based priority, on the other hand, aims at giving preference exactly to the patches containing image structures. However, in the case of digitized paintings, these structures are usually difficult to determine due to the painting technique (incomplete brushstrokes), scanning artefacts, etc. Moreover, undetected cracks can be interpreted as object boundaries, thus having high priority, which results in their continuation. For that reason, we test the GBCA method with two different definitions of priority.

The performance of these methods is evaluated only by visual inspection, as in the other papers on this application [Giakoumis 06,Solanki 09,Spagnolo 10]. Quantitative comparison is infeasible due to the unavailability of the ground truth data, i.e., we have no information on how the painting looked like in its original state, before the deterioration of wooden panels. On the other hand, the nature of the painting itself and the influence of the acquisition process of the digitized version (such as noise and scanning artefacts), make it very difficult to replicate the problem in a form of a suitable toy example on which the objective measurements could be performed.

By visually comparing the results of the four above mentioned methods, namely CAD, MBCA, GBCA-O and GBCA-C (see Fig. 6.8), we can see that all patch-based methods outperform the pixel-based CAD. This is also notable in the results in Figs. 6.10, 6.11 and 6.12. Although CAD performed sufficiently well for other case studies in literature, where the cracks were represented by thin lines, in our experiments this method was unable to reproduce texture and to fill in larger holes. This poor performance of CAD can be attributed to the special characteristics of the cracks in the Ghent Altarpiece, e.g., their relatively large width and whitish borders (see Section 6.1.2). Furthermore, the quality of the scans, i.e., the presence of noise and scanning

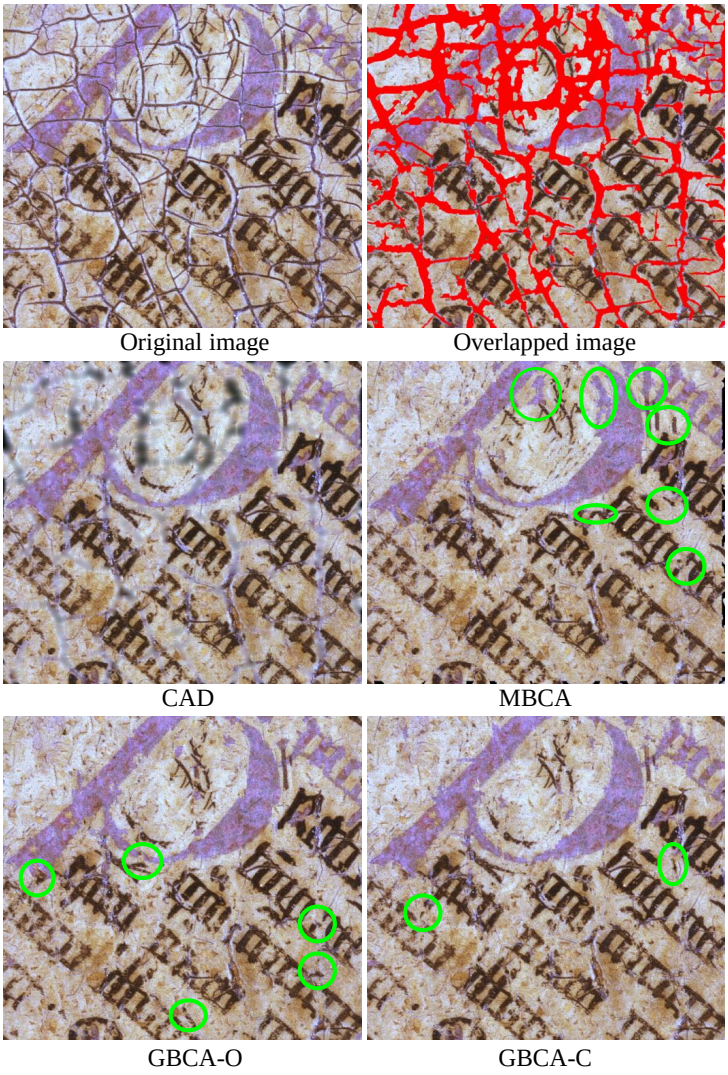


Figure 6.8: Comparison of crack inpainting results for the 750×825 part of the image in Fig. 6.7. From left to right and top to bottom: original image, original image overlapped with the combined dark and border crack map, result of CAD [Giakoumis 06], result of our MBCA method (Chapter 5), result of our GBCA-O method (Chapter 4), and result of our GBCA-C method. For our three methods, we used Gabor filters of 6 orientations and across 3 scales and 17×17 patches. Additionally, for MBCA, the rest of the parameters are the same as in Section 5.4.2, except $T_b = 0.03$. For GBCA-O and GBCA-C, we used division into 6×6 blocks and $r = 6$, and for GBCA-O, $T_{OE} = 10^{-3}$.

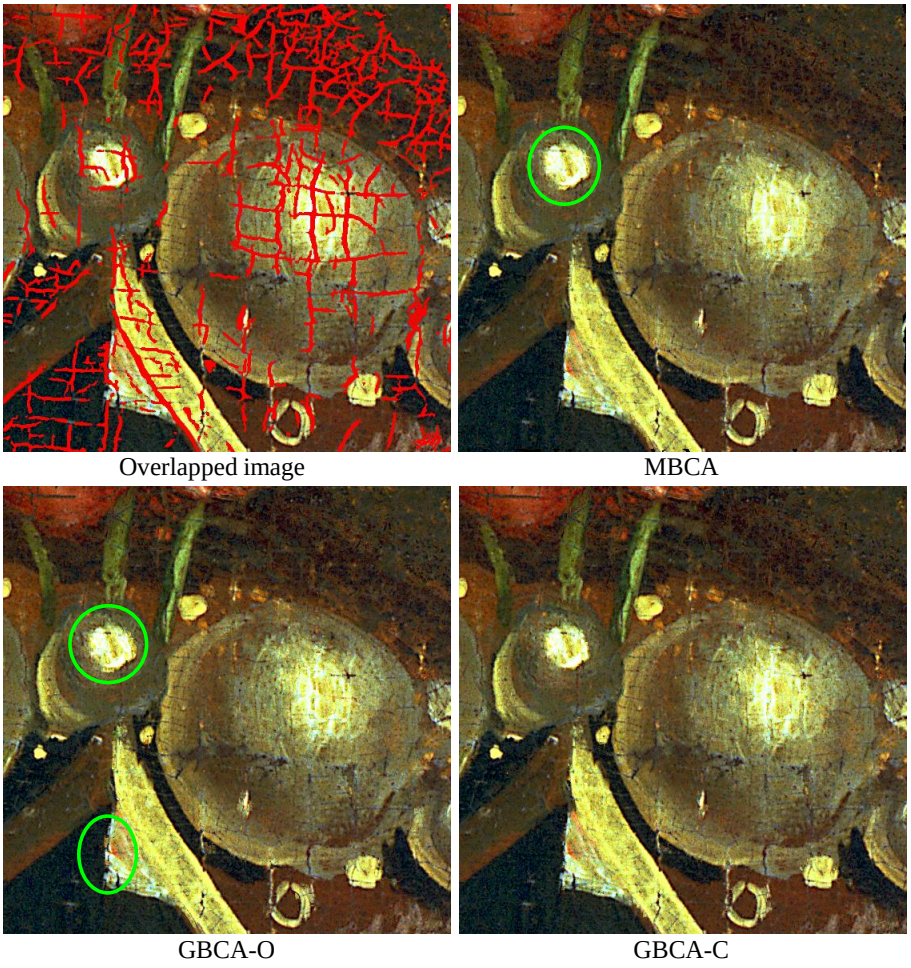


Figure 6.9: Comparison of crack inpainting results. From left to right and top to bottom: original image overlapped with the combined dark and bright crack map, result of our MBCA method, result of our GBCA-O method, and result of our GBCA-C method. We used Gabor filters of 6 orientations and across 3 scales and 11×11 patches. Additionally, for MBCA, the rest of the parameters are the same as in Section 5.4.2, except $T_b = 0.1$. For GBCA-O and GBCA-C, we used division into 4×4 blocks and $r = 6$, and for GBCA-O, $T_{OE} = 10^{-3}$.

artefacts, raises the need for better texture replication because diffusion-based methods produce blurry results.

In Figs. 6.8 and 6.9, we also compare our three proposed patch-based methods among each other. They all perform relatively well, but they still leave room for improvement when crack inpainting is considered. The complex MRF-based method, MBCA, performs similarly to the two simpler greedy ones,

although some artefacts are noticeable, especially in the result in Fig. 6.8 (see marked areas). Additionally, this method is about an order of a magnitude slower than our GBCA methods (see Section 5.4.2). In crack inpainting, the problem of the high computational load is especially aggravated due to the high resolution of the scans, making it impractical for the processing of larger areas. On the other hand, limiting the method to small areas can jeopardize finding the right matching patch. Therefore, we adopt the greedy patch-based method from Chapter 4, and improve it for crack inpainting. In particular, from now on we will use the GBCA-C method, with the priority as defined in Eq. (6.1), because it introduces the least number of artefacts (see Figs. 6.8 and 6.9). Comparison of the results of this method and CAD on a bigger image is shown in Fig. 6.10.

6.5 Combining dark and bright crack map

To improve the inpainting performance, some specifics of the problem need to be taken into account. In some cases, the presence of bright borders around the cracks (see Section 6.1.2) causes the missing crack regions to be filled with incorrect content and the positions of cracks to remain visible after inpainting (see the results on the left of Figs. 6.11 and 6.12). This is due to the fact that most inpainting algorithms fill in the missing (damaged) region based on pixel values from its immediate surroundings, which in this case are the aforementioned bright borders. Note in the second row on the left of Fig. 6.11 that CAD is especially sensitive to this problem, because it generally suffers from the introduction of blur. Often, the problem of the existence of bright borders is partially solved by using the bright crack map, which extends the dark crack map with the corresponding bright regions. Because this bright crack map also marks some of the bright borders, the benefit of using this map is evident in all cases: in the results on the right of Fig. 6.11, more cracks are detected and inpainted, causing a more pleasing visual appearance.

However, for some images this procedure might not be sufficient due to the width of the borders, as mentioned earlier in Section 6.3. In those cases, the inpainting pre-processing is used to obtain the map of crack border locations (see Fig. 6.7). The improvement of the inpainting results is shown on the right of Fig. 6.12, both for CAD and our patch-based GBCA-C method, in comparison with the results obtained using just the dark crack map shown on the left of Fig. 6.12. If the image also contains bright cracks, all three crack maps (dark, bright and border crack map) are combined together.

6.6 Segmentation-based candidate selection for crack inpainting

As we demonstrated so far, our patch-based GBCA-C method gives reasonably good visual results for most parts of the panels. However, the book of the

Figure 6.10: Comparison of crack inpainting results for the detail from the *God the Father* panel (see original image on the far right of Fig. 6.2). From top to bottom: original image overlapped with the combined dark and bright crack map, result of CAD, and result of our GBCA-C method (we used Gabor filters of 6 orientations and across 3 scales, 15×15 patches, division into 8×8 blocks, and $r = 6$).

Figure 6.11: Influence of bright borders on inpainting. Left: dark crack map as input. Right: combined dark and bright crack map as input. The first row shows the original image overlapped with crack maps. The second and the third row show inpainting results obtained with CAD and our GBCA-C method, respectively. For our GBCA-C method, we used Gabor filters of 6 orientations and across 3 scales, 17×17 patches, division into 6×4 blocks, and $r = 6$.

Annunciation to Mary panel is exceptionally difficult to process due to the width of the cracks, prominent scanning artefacts and imperfect brush strokes

(see Fig. 6.3). This causes some cracks to remain undetected and misguides the inpainting during the patch matching process. The first consequence is that we can get an inpainted image where small parts of letters appear erroneously in the background, and the other way around, parts of letters get "deleted", i.e., replaced by the background. The second consequence is that positions of cracks remain visible (see the result in the bottom right of Fig. 6.12). Exactly in the part of the panel containing the book, accurate inpainting is very important because of paleographical deciphering of the text (see Section 6.1.2 and later Section 6.6.3).

6.6.1 General idea

To further improve crack inpainting results, we propose a novel crack inpainting method that involves two contributions:

- the segmentation-based approach to patch candidate selection, and
- the approach to patch size adaptation.

Like our methods from Chapters 4 and 5, this method also aims at performing context-aware inpainting: the search for candidate patches is constrained to image regions, with the context well matching the context of the current target patch. However, in this method contextual information is adapted to the particular application. In the case of the image of the book from Fig. 6.3, we segment the image into background (page of the book) and foreground (letters). Such segmentation allows us to make a better distinction between these two important components of this image than our previously proposed contextual descriptors within fixed or even adaptive blocks. In particular, since letters are equally distributed across the image, all these blocks contain both letters and background, and thus block matching is not able to substantially guide the inpainting process (see Fig. 6.13).

Our proposed method for crack inpainting consists of three main steps:

1. **Exclusion of damaged pixels**

   Although we use the bright crack map and/or border crack map to deal with the problem of whitish borders around the cracks (see Section 6.5 and Fig. 6.12), some damaged pixels still remain. These pixels are either too distant from the crack, belong to the non-detected cracks, or appear in the source region not related to the cracks. Our idea is to detect these pixels within the current target patch based on their colour properties, and treat them as missing ones. Additionally, we do not use the patches from the source region containing damaged pixels as possible matches.

2. **Segment-constrained matching**

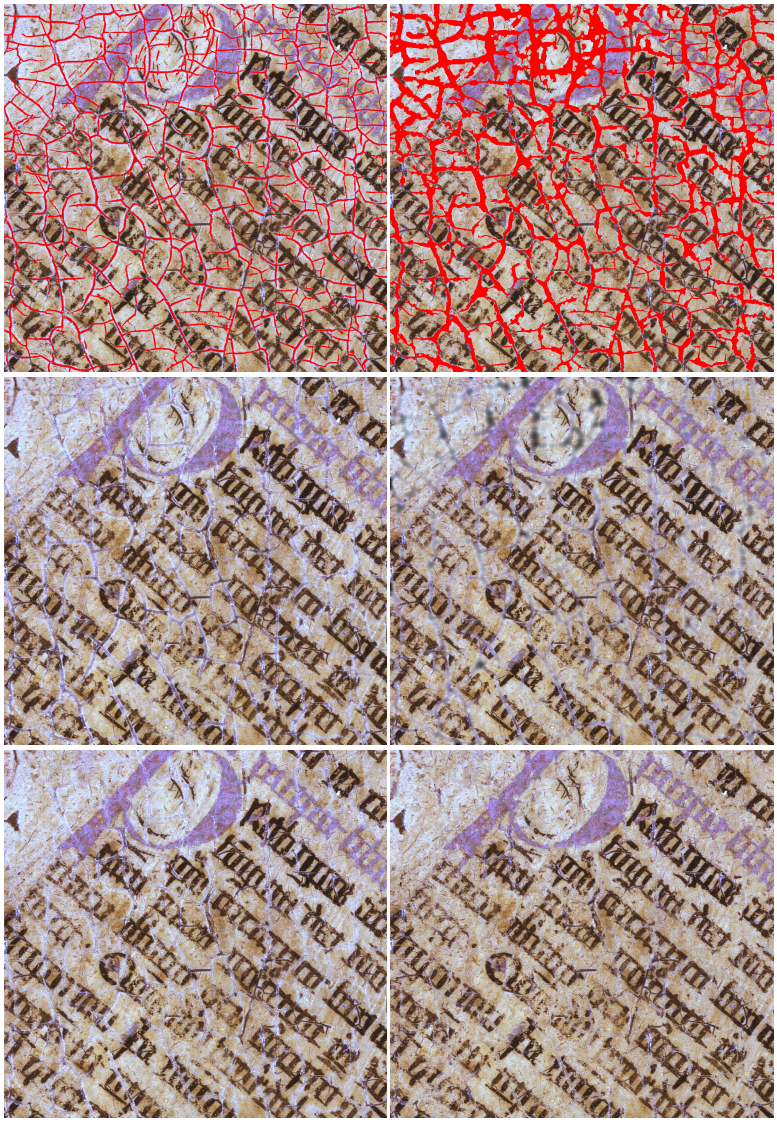


Figure 6.12: Influence of bright borders on inpainting. Left: dark crack map as input. Right: combined dark and border crack map as input. The first row shows the original image overlapped with crack maps. The second and the third row show inpainting results with CAD and our GBCA-C method, respectively. For our GBCA-C method, we used Gabor filters of 6 orientations and across 3 scales, 17×17 patches, division into 6×6 blocks, and $r = 6$.

In the results in the bottom row of Fig. 6.8 and in the bottom right of Fig. 6.12, it can be seen that patch-based inpainting occasionally intro-

duces some artefacts. This can happen because the known part of the target patch is not distinctive enough to find the right source patch. Another reason is that undetected cracks can be present in the known part of the target patch so that the matched source patch will probably contain a letter, since cracks and letters often have similar properties. In order to minimize these errors, we first segment the image in two classes: foreground (letters and undetected cracks) and background (page of the book). Based on this segmentation, we constrain the search for candidate patches, in the sense that if the target patch completely belongs to one segment, we constrain the search for candidate patches to that segment only. Therefore, this approach has very similar reasoning like our earlier proposed context-aware inpainting methods, but rather than using contextual descriptors within image regions, we explore the use of image segmentation.

3. **Adaptive patch size**

   Instead of using a fixed patch size, as most inpainting methods do, we adapt the patch size to the local context. As in the previous step of the algorithm, context is determined via image segmentation. Our idea is to gradually reduce the patch size while performing segment-constrained matching in order to localize better the patches within the segments.

Next, we present formally above described ideas within a novel crack inpainting method.

6.6.2 Proposed crack inpainting algorithm

Similarly to our GBCA inpainting method (Section 4.5.2, see also Chapters 3 and 4 for notations), we fill in the missing region iteratively, where in each iteration we search for the best-matching patch of the current target patch in the source region. Considering the aforementioned observations, we constrain this search based on two properties: 1) whether the candidate source patches contain damaged pixels (step 1 in Section 6.6.1), and 2) the context surrounding the current target patch (step 2 in Section 6.6.1). We make a distinction between "damaged" and "undamaged" pixels based on their values in the blue plane of the RGB representation of the image, denoted as g_B.[5] Therefore, we enforce the first constraint by creating the new "undamaged" source region $\Upsilon \subset \Phi$ as:

$$\Upsilon = \{p \in \Phi | g_B(p) \leq T_d\}, \tag{6.2}$$

where T_d is some threshold. Note that, by definition, the candidate source patch is completely inside the source region Υ, thus it cannot contain any damaged pixels.

[5]The same feature was used to detect the white borders in Section 6.3.

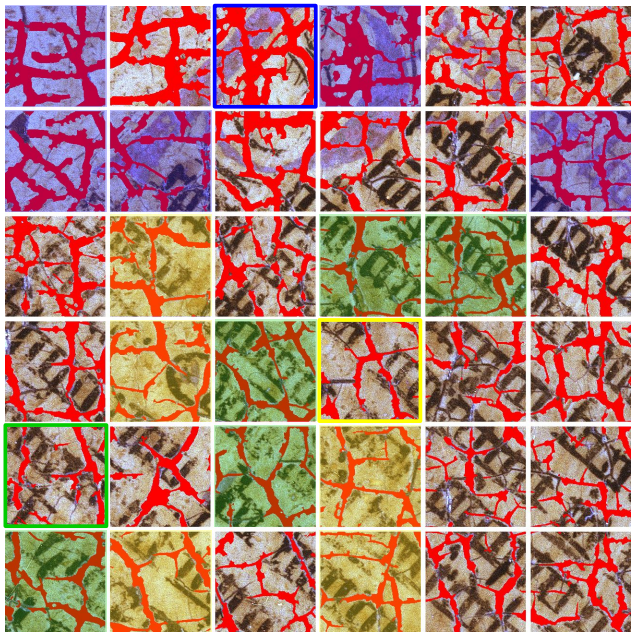
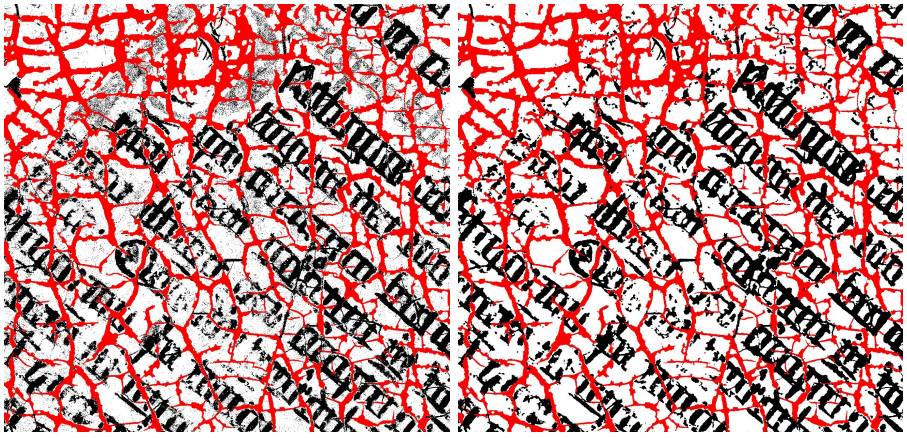


Figure 6.13: Division of the image of the book into 6×6 non-overlapping blocks. Block matches of the blocks in outlined squares are marked with the matching colour.



(a) **(b)**

Figure 6.14: Segmentation results for the part of the book from Fig. 6.3 (cracks detected as in Section 6.3 are marked in red, letters and undetected cracks in black, and background in white): (a) result of the K-means, and (b) result of the MRF-based segmentation.

We determine the context based on image segmentation into foreground and background. For segmentation, we use the MRF-based approach, similar to the one introduced previously in Section 2.5.3. The MRF is defined over image *pixels*, thus pixels represent MRF nodes, and the labels x_i assigned to nodes i represent the segments to which each pixel can belong. The pairwise potential is defined via the Potts model (Eq. (2.7)), while the local evidence is defined as

$$\phi(x_i, y_i) = \frac{1}{\sqrt{2\pi\sigma(x_i)^2}} \exp\Big(\frac{-(y_i - \mu(x_i))^2}{2\sigma(x_i)^2}\Big), \tag{6.3}$$

where $\mu(x_i)$ and $\sigma(x_i)$ are computed per each segment x_i as mean value and standard deviation of all the pixels that *initially* belong to the segment x_i, where the initial segmentation is obtained with the K-means algorithm [Duda 73]. y_i represents the observation, which is the measured pixel value at the node i. We used our neighbourhood-consensus message passing (NCMP) method for inference (Section 2.4). Fig. 6.14 shows how the MRF-based segmentation approach improves on the result of the K-means segmentation [Duda 73], which is noisy with a lot of misclassified isolated dots in the background. The MRF-based segmentation removes these isolated dots and yields more compact letters. In this way, better segmentation-based context-awareness can be achieved. Let $\Delta \subset I$ and $\Xi \subset I$ denote the background and the foreground segment, respectively, where $\Delta \cup \Xi = I$.

After creating the new source region and performing segmentation, we start the inpainting process. In each iteration t, we first find the target patch of the highest priority, whose central pixel $\hat{p}$ is computed like in Eq. (4.14), and the priority $R(p)$ is defined in Eq. (6.1). Within the known part of this patch, we detect the undamaged pixels as the ones whose value in the blue plane is lower or equal to the threshold T_d. The other known pixels of the current target patch are considered damaged and thus are treated as missing, i.e., as the pixels that need to be inpainted (step 1 in Section 6.6.1). In this way, we prevent the damaged pixels from misguiding the inpainting process. We can formally define the set of *current* known undamaged pixels as

$$\Upsilon^{(t)} = \{p \in I \setminus \Omega^{(t)} | g_B(p) \leq T_d\}, \tag{6.4}$$

where $\Omega^{(t)}$ denotes the *current* target (missing) region. We introduce the set $\Upsilon^{(t)}$ because we need to keep the undamaged source region Υ fixed throughout the algorithm. Note also that $\Upsilon \cup \Omega \neq I$, because we want to replace only the damaged pixels that are in the vicinity of the cracks and not elsewhere in the image. The threshold T_d is chosen high enough to allow sufficient number of candidate patches, while still detecting the artefacts around cracks. In particular, we chose a fixed threshold $T_d = 220$ by inspecting the histogram of manually marked damaged regions.

The next step is to find the best-matching patch of the current target patch based on its known *undamaged* pixels by performing segment-constrained

matching. The proposed segment-constrained matching enforces the following: when *all* the known undamaged pixels in the target patch belong to the background, we only accept source patches that belong completely to the background *and* the undamaged source region as candidate patches. Otherwise, the search is performed everywhere in the undamaged source region. We can define formally the central pixel of the best-matching patch of the current target patch centred at $\hat{p}$ as

$$\hat{q} = \begin{cases} \arg\min_{q \in (\Upsilon \cap \Delta)} \|\mathcal{T}_{\hat{p}} g - \mathcal{T}_q g\|^2_{(\Upsilon^{(t)} - \hat{p}) \cap \Psi}, & \text{if} \quad (\Psi + \hat{p}) \cap \Xi = \emptyset \\ \arg\min_{q \in \Upsilon} \|\mathcal{T}_{\hat{p}} g - \mathcal{T}_q g\|^2_{(\Upsilon^{(t)} - \hat{p}) \cap \Psi}, & \text{otherwise,} \end{cases} \tag{6.5}$$

using the notations and definitions introduced in Section 3.3. We could perform a similar procedure for the target patches belonging completely to the foreground, i.e., for which $(\Psi + \hat{p}) \cap \Delta = \emptyset$. However, some cracks that remain undetected by using the detection methods from Section 6.3 are also identified as foreground. This can result in unjustified insertion of letters and/or cracks (foreground) in the background. Therefore, if the target patch is not entirely in the background, we search through all possible candidates.

Finally, we perform the search for the best-matching patch by adapting the patch size according to the local context. We start from the maximal patch size and check if the target patch completely belongs to the background. If this is the case, we constrain the search to the background. If not, we reduce the patch size by half and repeat the same procedure. If even this smaller patch only partially belongs to the background, we search for the match of the target patch of the maximal size at all possible locations. Once the best match is found, we copy the corresponding pixels from this match to the locations of missing pixels.

6.6.3 Results

The effects of the proposed method from Section 6.6.2 are illustrated in Fig. 6.15(d). The result of the proposed method when using the fixed patch size instead of adaptive patch size is shown in Fig. 6.15(c). Comparing these two results, we can see that using adaptive patch size produces better result, with less artefacts in the background, meaning that the adaptive patch size approach can better locate target and source patches belonging to the background. Furthermore, some letters are better inpainted. In comparison with the result of our GBCA-C method in Fig. 6.15(b), the letters are better inpainted and the whole image contains less visually disturbing bright borders.

The result on the whole book is shown in Fig. 6.18. Since this image is large in size (4166×5206 pixels), we divided it into regions and performed inpainting in each region separately. Furthermore, we only inpainted the region of interest, i.e., the part of the image containing letters, leaving the rest of the image untouched. In [Cornelis 13], we showed that this result indeed improved legibility of the text in this book, as implied in Section 6.1.2. Crack inpainting

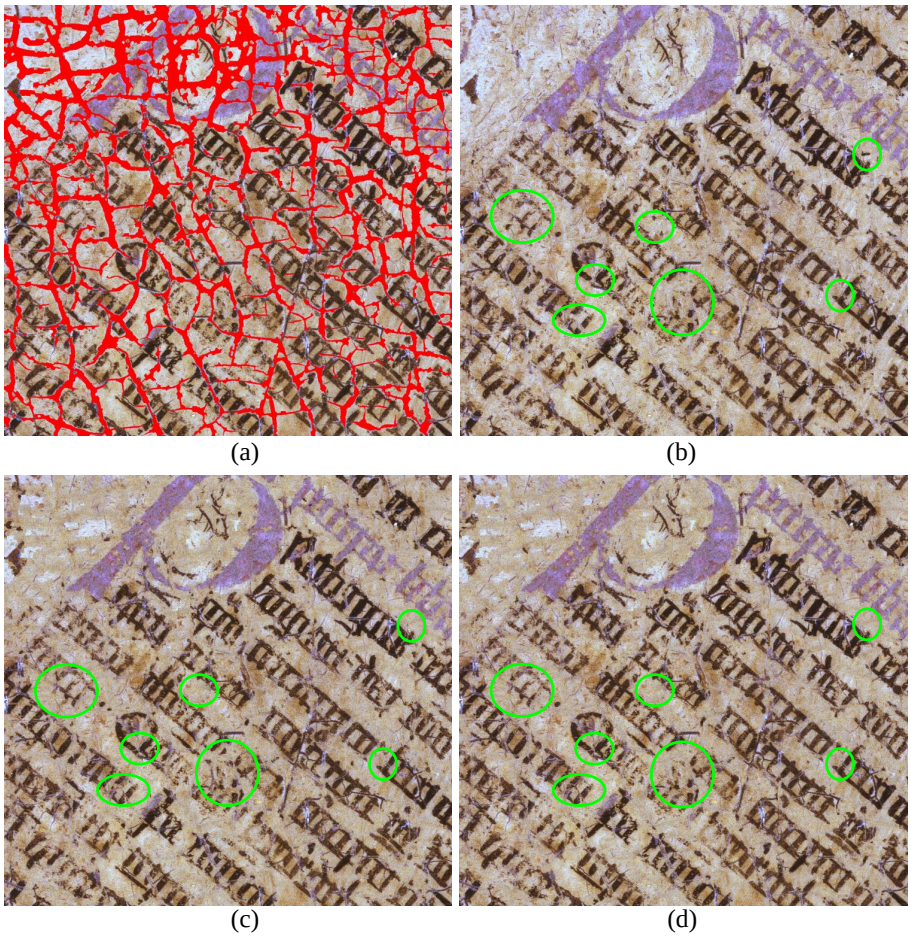


Figure 6.15: Comparison of inpainting results for the part of the book from Fig. 6.3: (a) original image overlapped with the combined dark and border crack map, (b) result of our GBCA-C method (see Section 6.4 and the caption of Fig. 6.12 for parameters), (c) result of the proposed crack inpainting method with fixed patch size, and (d) result of the proposed crack inpainting method with adaptive patch size.

enabled the deciphering of some additional word groups, although the text still cannot be read entirely. These deciphered word groups are: *hio dicta significata* (telling the message with mouth wide open), *de virtutibus d[ei]* (on the virtues of God), *in videndo* (the appearance of God). The former reading of *Prologus iste est ad* can be completed with the words *differentiam cognite dei*. Moreover, the paragraph mark on the upper left of the page should be read as *LXII* (62) rather than *VII* (7). All deciphered text fragments are related to the Annunciation, and can be found in Thomas of Aquino's *Summa Theologica* (written between 1266 and 1273). These first results provide a basis for further

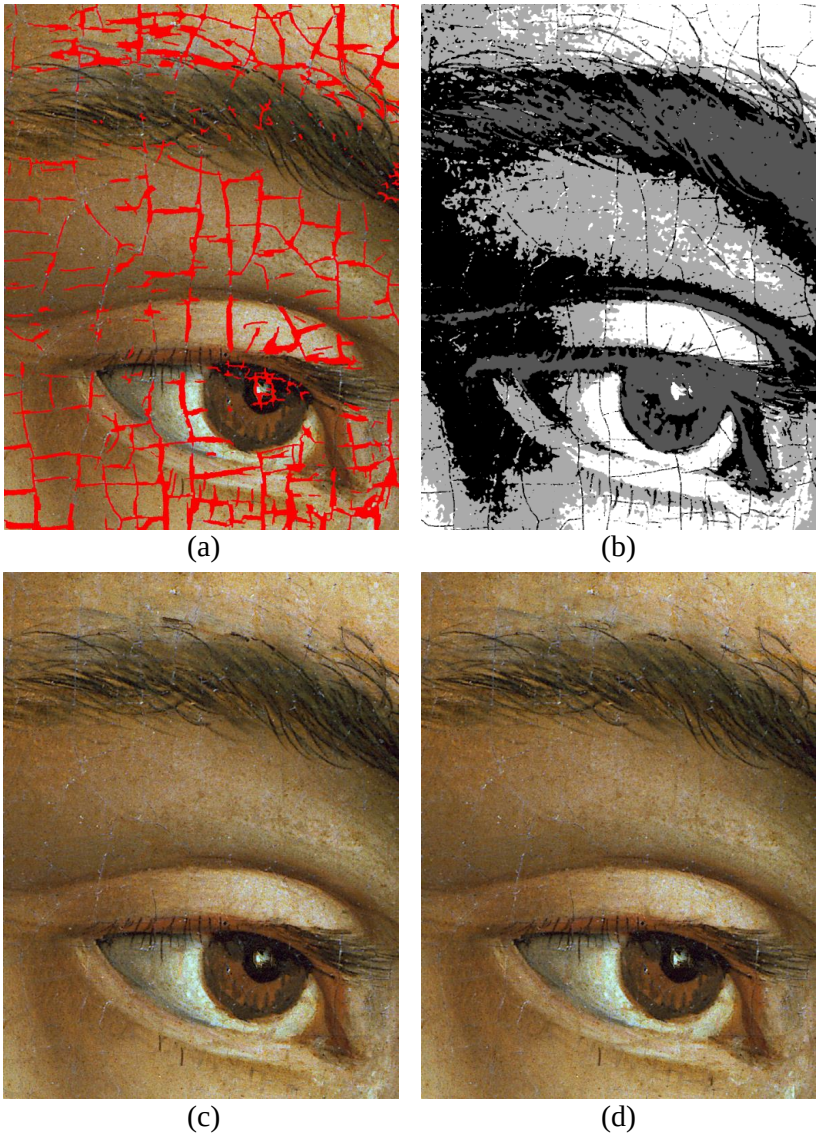


Figure 6.16: Comparison of inpainting results for Adam's Eye from Fig. 6.4: (a) original image overlapped with the combined dark and bright crack map, (b) result of the MRF-based segmentation, (c) result of our GBCA-C method, and (d) result of the proposed crack inpainting method with segment-constrained matching. For both methods, we used 15×15 patches, and for GBCA-C, we used Gabor filters of 6 orientations and across 3 scales, division into 8×6 blocks, and $r = 6$.

research into the iconographical implications of this text.

Note that our segment-constrained matching, i.e., the second step of

the complete method, can in principle also be used on more complex images, containing more than two segments, to limit the search to specific areas so that the computation time is reduced. However, in general, the improvement of the quality of the inpainting result compared to the block-based context-aware inpainting is minimal. Results are shown in Fig. 6.16: (c) shows the result of our GBCA-C method, and (d) shows the result of the proposed crack inpainting method with segment-constrained matching based on the segmentation shown in (b).

6.7 Conclusion

In this chapter, we addressed an interesting and challenging application of image inpainting for the virtual removal of cracks in old paintings. We first explored the use of our patch-based inpainting methods, proposed earlier in this thesis, for crack removal in the case study of the Ghent Altarpiece. While these methods were shown to outperform other related crack inpainting methods, they still left some room for improvement. The main contribution of this chapter is a novel patch-based inpainting method, which is specifically designed for virtual removal of cracks in old paintings. The proposed method performs context-aware inpainting, but rather than describing the context with texture (and colour) features within image blocks of fixed or adaptive sizes, like in our previously proposed methods, we explore the use of image segmentation for context description. The idea is to constrain the search for candidate patches to appropriate *segments* of the image, while also adapting the patch size according to the context. Additionally, the method is capable of dealing with some specific problems of the cracks in the Ghent Altarpiece, such as the existence of whitish crack borders.

A special attention was devoted to inpainting the image of the book in the *Annunciation to Mary* panel. The text in the book is of special interest to art historians, because its paleographical deciphering can give new insights to the meaning of this painting. We showed that the proposed crack inpainting method produces more accurate results, in the sense that the positions of cracks are not visible, letters of the text are better preserved and minimal artefacts are introduced in the background. In this way, we were able to improve the legibility of the text and contribute to the art historical analysis.

This work resulted in one journal paper as the second author [Cornelis 13], one book chapter as a co-author [Pižurica 13], two conference publications [Ružić 11a, Ružić 13a], and two abstracts were presented in international conferences [Ružić 10, Cornelis 11]. These results also attracted the attention of wider audience, resulting in the article in the popular Belgian and Dutch science EOS Magazine (June, 2012), in the Flemish newspapers De Standaard[6] (March 27, 2013) and Het Nieuwsblad[7] (March 27, 2013), in the VRT's online

[6] http://www.standaard.be/cnt/dmf20130327_00520016
[7] http://www.nieuwsblad.be/article/detail.aspx?articleid=DMF20130326_00519077

cultural magazine Cobra[8] (March 27, 2013), and in the Schamper magazine of Ghent University[9] (April 15, 2013). They were also mentioned in the two Press Releases of Ghent University: on the occasion of the official start of the physical restoration of the Ghent Altarpiece (September 7, 2012)[10] and about the Ghent University research related to Lam Gods (March 26, 2013)[11]. Finally, this research was presented on several invited talks at prestigious events: Het Lam Gods Series of Lectures, Provinciaal Administratief Centrum P.A.C. Ghent[12] (November, 2012) and TEDxGent[13], Aula, Ghent (June, 2012).

[8] http://www.cobra.be/cm/cobra/kunsten/1.1586571
[9] http://www.schamper.ugent.be/527/op-zoek-naar-het-lam-gods
[10] Persbericht: "Een traditie van innovatief onderzoek van het Lam Gods aan de UGent"
[11] http://www.ugent.be/nl/actueel/nieuws/lam-gods.htm
[12] http://www.csct.ugent.be/
[13] http://www.youtube.com/watch?v=kvVb5NG6TLk

Figure 6.17: Original image of the book from the *Annunciation to Mary* panel.

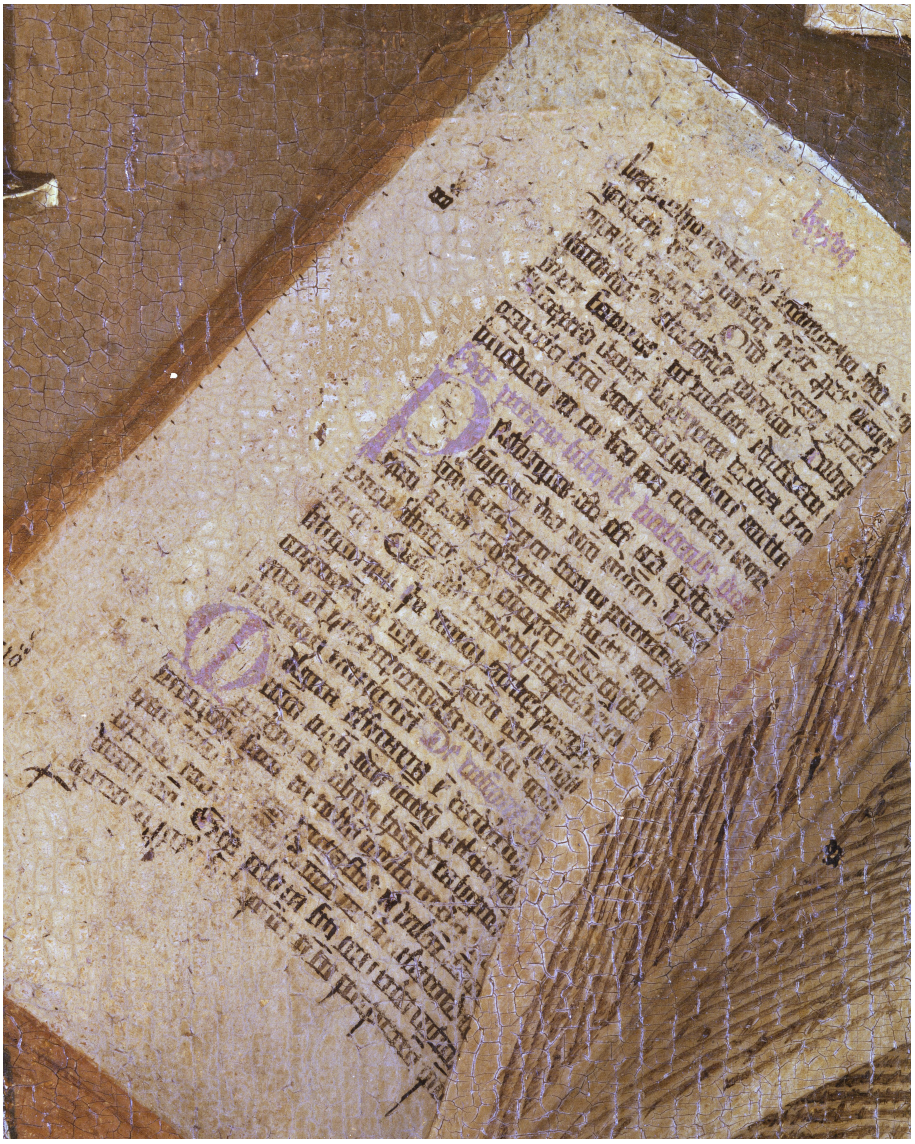


Figure 6.18: Result of the proposed crack inpainting method on the image of the book from the *Annunciation to Mary* panel (Fig. 6.17).

7

Conclusion

Even with the advance of digital imaging acquisition devices, acquired images still contain degradations in terms of resolution, noise, artefacts, etc. These degradations can be caused by imperfect acquisition devices (e.g., due to physical limitations and cost restrictions), or by ageing of the material to be digitized (e.g., scratches in scanned old photographs or artefacts such as cracks and stains in digitized artwork). Digital post-processing techniques are an inexpensive and often the only way to remove these degradations in order to improve the quality of digital images. Moreover, this facilitates the analysis of image content in applications like surveillance, forensics, satellite, medical imaging and art historical analysis. Some of these digital post-processing techniques can also be used for image editing, e.g., removing unwanted elements from images, like stamped date, watermarks, text, logos, or even the whole objects.

In this thesis, we developed digital post-processing techniques to restore and edit images after acquisition, which are based on Markov random field (MRF) models and patch representations. We focus on super-resolution (SR) and inpainting application. In this chapter, we first review our main contributions and then outline a few directions for future research.

7.1 Review of our contributions

MRFs are widely used in image processing and computer vision problems because they provide a convenient and consistent way of modelling contextual constraints, like spatial correlations among image pixels and spatial consistency among other image entities. In particular, MRFs are able to model global image context in terms of local interactions, which makes this model elegant and computationally tractable. MRFs are often used as a prior in problems that involve Bayesian inference, like maximum a posteriori (MAP) estimation, where the goal is to estimate some unknown image attributes from the available image data, which are incomplete or degraded. In Chapter 2, we developed a novel suboptimal inference method for MAP estimation with the MRF prior, which is based on message passing. We called this method neighbourhood-consensus message passing (NCMP) since a joint "consensus"

message is sent from the specified neighbourhood to the central node, rather than relying on pairwise interaction only. We showed that the proposed method can be considered as a generalization of the iterated conditional expectations (ICE) inference algorithm. Additionally, we developed a simplified version of NCMP, called weighted iterated conditional modes (WICM), which is especially efficient when working with large neighbourhoods. Results on different example applications showed the potentials of the proposed methods. In particular, NCMP always improved over the popular greedy method of iterated conditional modes (ICM) with only mild increase in complexity. Moreover, NCMP in most of the practical labelling applications yielded similar results as the state-of-the-art loopy belief propagation (LBP), while being much less complex.

The proposed NCMP is generally applicable to a wide range of problems, which allowed us to employ it as an inference engine for MRF patch-based models for SR and inpainting. In Chapter 3, we addressed the SR problem using contextual modelling: the unknown high-resolution (HR) patches are estimated based on the agreement with the available low-resolution (LR) versions and based on the prior knowledge about spatial consistency among neighbouring HR patches, encoded by the MRF prior. Other methods that use these types of models typically employ an external database from which the HR patches are extracted. We proposed a novel single-image patch-based SR method, where we exploited the self-similarity of image patches in natural images across different scales, thus the HR patches are taken from the input image itself. Visual and quantitative comparison of results, in terms of root mean square error (RMSE) and structure similarity index (SSIM), showed that our method greatly outperforms standard techniques, while being visually better or comparable with state-of-the-art techniques. Results could be further evaluated in some computer vision tasks, e.g., for feature detection, but this was not investigated within the scope of this research.

In Chapters 4, 5 and 6, we also exploited image self-similarity, but within the same scale, for the problem of image inpainting. Here, the idea is to search for well-matching candidate patches of the patch to be inpainted in the known part of the image. The main contribution of Chapter 4 is a novel context-aware patch selection approach, which reduces the number of candidate patches and chooses them in such a way that they better fit the surrounding context. We represented context within blocks of fixed size using contextual descriptors in the form of combined texture and colour features. Texture features were obtained by filtering the image with the bank of Gabor filters at multiple scales and orientations (the so-called multi-channel filtering). Comparison of these contextual descriptors enabled us to find regions of similar context in the image, as we demonstrated with intermediate results. Such context-aware approach is general, and thus can be applied with any patch-based inpainting algorithm. In Chapter 4, we employed the proposed approach within a novel greedy block-based context-aware (GBCA) inpainting method, whose additional contribution is a novel orientation-based priority, which deter-

mines the filling order of the missing region. This definition of priority is based on contour features, which were also obtained by multi-channel filtering. We demonstrated that the meaningful constrained search for patches yields better inpainting result in less time than the exhaustive search (i.e., searching over the whole known part of the image). Furthermore, our results were visually better or comparable with state-of-the-art methods.

In Chapter 5, we further evolved our work on context-aware patch-based inpainting. We introduced a novel context representation within blocks of adaptive sizes using contextual descriptors in the form of normalized texton histograms. For the division of the image into blocks of adaptive sizes, we proposed a novel top-down splitting procedure, which is also based on contextual descriptors. We applied this improved context-aware approach within a novel MRF block-based context-aware (MBCA) inpainting method. In this way, the speed and the performance of the so-called global patch-based image inpainting with the MRF prior is improved. To solve the inference problem in this MRF model, we proposed a simple and efficient method, which builds upon our NCMP inference method to make it suitable for global inpainting problem with large number of labels. We evaluated the proposed method on two example applications: scratch and text removal and object removal. Results demonstrate the benefits of our approach in comparison with state-of-the-art methods in terms of quality and additionally, in comparison with another MRF-based method, in terms of speed.

In Chapter 6, we applied the developed inpainting methods on the interesting and challenging application of virtual removal of cracks in old paintings. As a case study, we used the digitized versions of the *Adoration of the Mystic Lamb*, also known as the *Ghent Altarpiece*. We showed how the proposed methods outperform related crack inpainting methods. However, they still leave some room for improvement due to the particularities of cracks in this painting, especially for the image of the book in the *Annunciation to Mary* panel. The text in this book is of special interest to art historians, because its paleographical deciphering can give new insights to the meaning of this painting. Therefore, we introduced a novel patch-based crack inpainting method, where the idea is to use image segmentation for context description. In particular, we constrained the search for candidate patches to appropriate segments of the image, while also adapting the patch size according to the context. We showed that the proposed crack inpainting method produces more accurate results, in the sense that the positions of cracks are not visible, letters of the text are better preserved and minimal artefacts are introduced in the background. In this way, we were able to improve the legibility of the text. In particular, some additional word groups were deciphered, although the text still cannot be read entirely. These first results provide a basis for further research into the iconographical implications of this text.

7.2 Future research

The research presented in this thesis opens several directions for future work, especially in the domain of image (and video) inpainting. In particular, it would be interesting to explore three topics: automatic evaluation of parameters, quality assessment techniques and application of inpainting for disocclusion filling in virtual view synthesis.

In the inpainting methods we developed within this research, we used several parameters, which we evaluated experimentally, i.e., we ran experiments for different values of parameters and chose the one that yielded the best result. Perhaps the most notable parameter is the patch size. As we discussed in Section 4.6, the choice of the patch size is a trade-off between capturing important structures or texture elements in the image and the ability to find good matches. Several methods in literature consider using adaptive patch size, where the main idea is to choose small patches for regions with high frequencies (i.e., textured and structured regions) and big patches for regions with low frequencies (i.e., flat areas). Our algorithms, on the other hand, use fixed patch size for all positions (except the crack inpainting method in Chapter 6). Therefore, in future work, two options can be explored: 1) using ideas about adaptive patch size to find an optimal fixed patch size depending on the image content and the size of the known region, and 2) extending our methods to the use of adaptive patch sizes. Another option would be to also consider using adaptive shapes, which are determined based on the image content. Regarding the proposed context-aware approach, we already introduced several improvements in Chapter 5, which resulted in less parameters and easier parameter optimization. However, still some dependencies on parameters remain, e.g., how to choose the optimal number of textons and consequently, the block similarity threshold, based on the given number of filters in the filter bank. Finding dependencies between these parameters and their dependency on the image content is a difficult and challenging task.

Another interesting question is how to compare inpainting results of different methods. Most of the time, comparison is performed visually, because of the absence of ground truth images in applications like object removal, scratch removal from old photographs and crack removal from digitized paintings. For artificially added artefacts, some quantitative measure can be computed, e.g., peak signal-to-noise ratio (see Section 5.4.1). However, a good result of image inpainting does not necessarily need to be as close as possible to the ground truth. Rather, inpainting should be performed in such a way that it is not noticeable to an observer that the image has been altered. In this respect, interesting research topic would be to develop quality assessment techniques for image inpainting, e.g., by exploring techniques from image forensics and/or performing human observer studies.

With the growing popularity of 3D television (3DTV) and free viewpoint television (FTV), image inpainting has found an important application for disocclusion filling in virtual view synthesis. Disocclusions are the missing areas to the left or right of the foreground objects, which were occluded in

the original view and become visible in the generated views. Removing these holes becomes demanding when 3D information is available in video-plus-depth format, which consists of the colour values of the central view and the depth map, or stereo (or multi-view) format during view extrapolation, i.e., synthesizing views outside the baseline. Our patch-based inpainting methods can be extended for this specific application by exploiting depth and temporal information and information from other views. Furthermore, certain simplifications of the algorithm can be made, e.g., feature reduction, fast approximate patch search, simpler approach for filling flat areas, etc. Finally, the algorithms can be implemented on the Graphical Processing Unit (GPU), in order to enable them to run in real time. This requires parallel processing and certain parts of the proposed algorithms can be executed in parallel, e.g., patch search and priority computation. Furthermore, inpainting of different areas of the missing region could be performed in parallel. These areas could be the blocks in our context-aware approach, although within the block the filling order should be imposed. Another option is to first continue important image structures inside the missing region, which then divide the missing region in different segments that can be processed in parallel. This research will be conducted within the iMinds ICON ASPRO+ project.

A

Texture and contour features

In this appendix, we give a theoretical background of texture and contour features, which we use as inpainting tools for our inpainting methods introduced in Chapters 4 and 5.

A.1 Extraction of texture features

Extraction of texture features has been widely studied in different image processing and computer vision tasks, such as automated inspection [Conners 83, Jain 90, Orjuela 13], medical image analysis [Chen 89], remote sensing [Rignot 90, Poggi 05], texture and image segmentation [Malik 01, Puzicha 97, Scarpa 09, Arbelaez 11], image retrieval [Puzicha 97], object detection [Rikert 99, Torralba 10], scene classification [Oliva 01], texture classification [Varma 03], surface recognition [Leung 99, Leung 01], analysis of paintings [van der Maaten 10], etc. There are many approaches on how to extract these features, and they can be categorized into statistical, geometrical, model-based and signal processing methods [Tuceryan 98].

Statistical approaches measure the spatial distribution of grey-level values [Tuceryan 98], and they include, e.g., autocorrelation features, co-occurrence matrices [Haralick 73] and sum and difference histograms [Unser 86]. Geometrical methods first identify "texture elements" or primitives, and then extract texture features either as some statistics of these primitives [Tuceryan 90] or their spatial arrangement, like in a very popular technique called local binary patterns [Ojala 96]. Model-based methods use stochastic or generative models (e.g., different kinds of MRF models [Chellappa 85, Andrey 98, Poggi 05, Scarpa 09]) to both describe and synthesize texture. The estimated model parameters are used as texture features. Finally, signal processing methods [Unser 90, Jain 91] extract a set of features from *filtered* images, which are then applied for classification or segmentation.

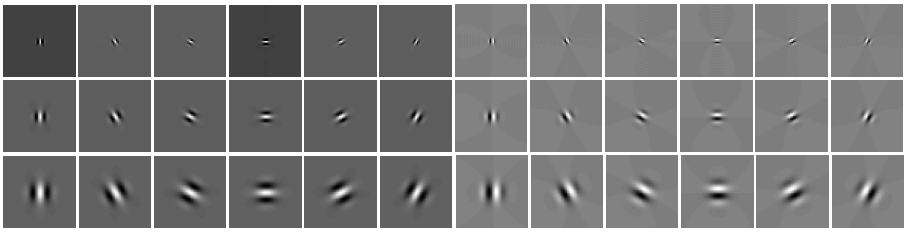
Among signal processing methods, and texture feature extraction

methods in general, *multi-channel filtering* is among the most popular ones. This approach analyses filter outputs of an image obtained by convolving an image with a bank of linear spatial filters at various orientations and scales. The inspiration for this approach comes from psychophysical studies of the human visual system [Hubel 62, Campbell 68, Devalois 82]. The study in [Campbell 68] suggested that the visual system decomposes an image into filtered images of various frequencies and orientations, while in [Devalois 82], it was shown that the receptive fields of simple cells in the visual cortex of some mammals are tuned to narrow ranges of frequency and orientation. Therefore, the multi-channel filtering approach models very well the processing of visual information in the early stages of the visual system.

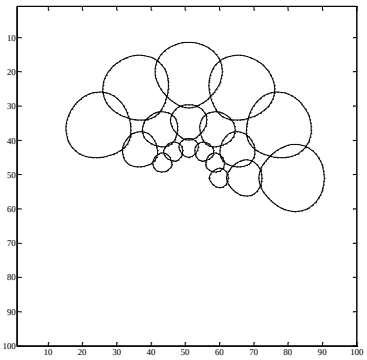
Note that some methods combine different approaches to extract texture features. Most often, they perform statistical analysis of filter responses in order to learn their *joint* distribution. In this way, dimensionality reduction is achieved and the influence of all filters at the points of interest is modelled, as opposed to independent distributions for each filter [Varma 03]. This joint distribution is represented by clusters [Leung 99, Rikert 99, Malik 01, Varma 05] or histograms [Zhu 98, Portilla 00, Konishi 00], and it can be used as a texture model for texture classification, synthesis or segmentation. We will visit the clustering-based method in more detail in Section A.5.

As mentioned earlier, multi-channel filtering is a widely used approach for feature extraction. Furthermore, it is used in image segmentation [Malik 01, Puzicha 97, Arbelaez 11], object detection [Torralba 10] and scene classification [Oliva 01], which were the motivation for our work, thus we will focus on this approach in this thesis.

Some of the filters that have been used for multi-channel filtering include Gaussian derivatives [Young 85], steerable pyramids [Simoncelli 92], Gabor filters [Gabor 46, Daugman 85], wavelets [Daubechies 92, Meyer 95, Mallat 09], and complex wavelets [Kingsbury 01, Selesnick 05]. Gabor filters represent a good model due to their optimal joint localization in both spatial and spatial-frequency domains [Daugman 85]. Furthermore, the studies in [Randen 99, Chen 99] showed that they outperform other filtering methods, such as ring/wedge filter, spatial filter, eigenfilter and wavelet transform. Finally, filters with very similar receptive fields as Gabor filters are obtained by performing different analysis on natural images, such as independent components analysis (ICA) [Hyvärinen 09], dictionary learning via sparse coding [Olshausen 97] or by training certain models based on MRFs [Osindero 06], which suggests that Gabor-like filters form the basis of natural images. Representation of an image via Gabor filter responses has been proven as very effective for texture analysis and many related applications, e.g., texture segmentation [Bovik 90, Jain 91, Weldon 96], object detection [Jain 97, Torralba 10], scene classification [Oliva 01], etc.



(a)



(b)

Figure A.1: (a) Illustration of Gabor filters across six orientations and three scales in spatial domain. Even filters (the first six columns) represent the real part of the complex Gabor filters, while odd filters (the other six column) represent the imaginary part. (b) Tiling of the frequency domain obtained by the filters from (a).

A.2 Multi-channel filtering using Gabor filters

A 2D complex Gabor filter (in spatial domain) consists of a complex sinusoid plane wave of a certain frequency and orientation modulated by a Gaussian envelope [Gabor 46]:

$$\mathcal{G}_{\eta,\tau,\sigma}(u,v) = \exp\Big(-\frac{u'^2+\vartheta v'^2}{2\sigma^2}\Big)\exp\Big(i\Big(2\pi\tau u'+\varphi\Big)\Big) \tag{A.1}$$
$$u' = u\cos\eta + v\sin\eta$$
$$v' = -u\sin\eta + v\cos\eta.$$

σ is the standard deviation of the Gaussian envelope, which determines the effective size of the surrounding of a pixel in which weighted summation takes place. ϑ, called the spatial aspect ratio, determines the eccentricity of the Gaussian, and it is kept constant (usually $\vartheta = 1$). τ defines the spatial frequency of the complex sinusoid, while $1/\tau$ is the wavelength. Then $\varsigma = \sigma\tau$ determines the spatial frequency bandwidth of the Gabor filter, i.e., the scale. Angle $\eta \in [0, \pi)$ specifies the filter orientation, i.e., the orientation of the normal to the parallel

stripes of the sinusoid. Finally, φ is the phase offset. A complex Gabor filter has a real and an imaginary component, which are conveniently in quadrature, i.e., out of phase by 90 degrees.

During multi-channel filtering, an image $\mathcal{I}$, as a continuous signal, is convolved with the bank of such filters at multiple orientations and scales. As a result, $\mathcal{I}$ is represented by a set of filtered images $\mathcal{F}_{\eta,\tau,\sigma}(u,v) = (\mathcal{I} * \mathcal{G}_{\eta,\tau,\sigma})(u,v)$, where $*$ is the convolution operator.

Since we are working with digital images, we will observe all the above operations in discrete space. Therefore, the image is defined over a lattice I, where positions on this lattice are represented by a single index $p \in I$ (assuming raster-scan order). The Gabor filters we use in this thesis are also discrete and they vary by scale ς and orientation η, while the rest of the parameters are kept constant. Therefore, we denote the discrete Gabor filter as $G_{\eta,\varsigma}$. Then the total number of complex filters in the Gabor filter bank is $N_f = N_\varsigma N_\eta$, where N_ς is the number of scales and N_ς is the number of orientations. If we consider the real and the imaginary parts of the filters separately, the total number of filters is $2 \times N_f$. Fig. A.1a illustrates 36 real and imaginary Gabor filters in spatial domain, corresponding to the complex filters of 6 orientations and across 3 scales, total of $N_f = 18$ complex filters. Fig. A.1b shows the obtained tiling of the frequency space. Finally, $\mathbf{f}(p)$ denotes an N_f-dimensional vector of *complex* filter outputs at the pixel p. This vector actually characterizes the image patch centred at p by a set of values at that pixel.

A.3 Texture features as averaged filter outputs

The vectors of filter responses can be used in applications like texture segmentation and classification to eventually describe the whole texture. It is also possible to use some other pixel-wise texture features extracted from these vectors (see [Clausi 00] for an overview and comparison). On the other hand, in some applications texture features can be used to describe an image *region*. For example, texture features were used for scene description by obtaining global image representation, rather than dividing a scene into objects [Oliva 01]. This representation is called a *gist* of an image, and it is based on computing statistics of low-level image features over fixed image regions, which divide the image into square non-overlapping blocks.

Let G_n denote one filter from the bank of complex Gabor filters, where the index $n = 1, \ldots, N_f$ represents the combination of filter orientation η and scale ς. Let $B + l$ denote a set of pixel position in a square block centred at $l \in \Theta$. The set Θ is determined by the division of the image into square non-overlapping blocks, and n_b is the total number of blocks. Then, the gist descriptor is the vector of features $\mathbf{d}$, where each feature is computed in the following way. First, the luminance channel of the image is filtered with each Gabor filter G_n from the filter bank, obtaining thus complex filter responses $f_n(p)$, $\forall p \in I$. Then, the magnitudes of these complex responses are averaged within each block $B + l$, i.e.,

$$d_n^{(l)} = \frac{1}{\#(B+l)} \sum_{p \in (B+l)} |f_n(p)|, \quad n = 1, \ldots, N_f. \tag{A.2}$$

Therefore, $\mathbf{d}^{(l)}$ is the feature vector per block whose dimensionality is N_f, while the gist of the whole image is $\mathbf{d} = (\mathbf{d}_1, \ldots, \mathbf{d}_{n_b})$. Gist as a scene descriptor was used for scene classification [Oliva 01], object detection and localization [Torralba 10], and scene completion using millions of photographs [Hays 08].[1]

A.4 Contour features

Contour features are mostly used in contour-based approaches for image segmentation, e.g., [Williams 95, Malik 01, Arbelaez 11]. The first step of these algorithms is often edge detection, which was in early approaches performed based on local measurements. Such edge detection algorithms include Sobel operator [Duda 73], zero crossings [Marr 80], Canny detector [Canny 86], etc. However, these approaches model brightness edges as step edges, which is insufficient for natural images due to phenomena such as mutual illumination, shading, depth or orientation discontinuities. Consequently, image edges require a richer description as a combination of steps, peaks and roof profiles [Malik 01]. Such a description can be obtained by analysing the response of an image to multi-channel filtering via the so-called *oriented energy* approach [Morrone 87, Perona 90]. This approach can be used to detect and localize these composite edges by employing quadrature pairs of even and odd symmetric filters. The obtained contour features are called *dominant orientation* and *oriented energy*.

Let $f_{\theta,\varsigma}^{even}(p)$ and $f_{\theta,\varsigma}^{odd}(p)$ denote filter responses to some even and odd symmetric filters, respectively, of orientation η and scale ς. Filter of orientation η will detect contours of orientation θ, which is orthogonal to the filter orientation (e.g., horizontally oriented filter with $\eta = 0°$ detects vertical contours, i.e., contours with $\theta = 90°$). Since for the analysis of contour features it is important to emphasize contour orientation, we use index θ to denote filter responses. The oriented energy at pixel p at scale ς and orientation θ is defined as [Perona 90]

$$OE_\theta(p) = \left(f_{\theta,\varsigma}^{even}(p)\right)^2 + \left(f_{\theta,\varsigma}^{odd}(p)\right)^2. \tag{A.3}$$

Then the dominant orientation at pixel p at scale ς is [Perona 90]

$$\theta^*(p) = \arg\max_\theta OE_\theta(p), \tag{A.4}$$

[1] In [Torralba 10], it is stated that gist is computed by averaging output *energy*, i.e., $|f_n(p)|^2$. However, the source code provided by the same authors, available at http://people.csail.mit.edu/torralba/code/spatialenvelope/, suggests that the magnitude of filter response is used. Since we use this code for all our Gabor filtering computations, we define gist according to the implementation.

and it represents the orientation of the contour at the pixel, while its corresponding oriented energy $OE_{\theta^*}(p)$ measures the contour's strength.

To better localize the contour, oriented non-maximal suppression [Canny 86] is used in the following manner. Let us denote as $p_1 \in I$ and $p_2 \in I$ the neighbouring pixels of p, which lie on the line orthogonal to the dominant orientation $\theta^*(p)$. $OE_{\theta^*}(p)$ is compared to the values $OE_{\theta^*}(p_1)$ and $OE_{\theta^*}(p_2)$. Then the final *oriented energy* is

$$OE^*(p) = \begin{cases} OE_{\theta^*}(p), & \text{if} \quad OE_{\theta^*}(p) \geq OE_{\theta^*}(p_1) \wedge OE_{\theta^*}(p) \geq OE_{\theta^*}(p_2) \\ 0, & \text{otherwise.} \end{cases} \tag{A.5}$$

Contour features are calculated per scale, and to obtain one feature per pixel, one can simply take the maximum over all scales [Malik 01].

Originally, in [Perona 90], even and odd symmetric Gaussian derivatives were used as filters. However, the analysis of features can be equivalently performed for Gabor filters [Malik 01].

A.5 Texton histograms

A.5.1 What are textons?

The term *texton* was first introduced by Julesz [Julesz 81] to represent the putative units of pre-attentive human texture perception. He qualitatively described them as simple binary line segment stimuli, such as elongated blobs, bars and crosses. He defined them by conducting experiments, where human subjects were asked to detect the target element among a number of distracting elements in the background, and then their response time was measured. Later it was shown that perceptual textons could be adapted through training [Karni 91]. However, these early psychophysical studies of textons relied on artificial texture patterns, thus lacked a mathematical definition for natural images.

Malik et al. revisited the concept of textons for the purpose of image segmentation in [Malik 99, Malik 01] (2D textons) and surface recognition [Leung 99, Leung 01] (3D textons). They presented a discriminative method for computing textons from grey-level images based on K-means clustering [Duda 73] of outputs of linear oriented Gaussian derivative filters. In this way, textons represent the prototypes of filter responses corresponding to local image structures, like edges, bars, corners, etc. 2D textons were also used for image segmentation in [Martin 04, Arbelaez 11], while 3D textons have been extended to textures with lighting variations and texture surface rendering [Liu 01, Cula 01, Varma 02, Varma 05].

An interesting approach is also to use directly raw pixel values to generate textons [Varma 03, van der Maaten 10], thus avoiding filter banks all together. In that case, feature vectors that are clustered are vectorized patches

surrounding each location in the image instead of vectors of filter outputs at those locations. Using pixel values directly was motivated by the success of texture synthesis algorithms [Efros 99, Zalesny 01, Wei 00], which use local pixel neighbourhood under an MRF assumption, in the sense that the pixel is conditioned on its neighbours (see Chapter 2). In [Varma 03], it was shown that this approach outperforms the filter bank-based approaches [Leung 01, Varma 02] for the purpose of texture classification. Later, this type of textons was also used for image segmentation [Schroff 06] and analysis of paintings [van der Maaten 10].

However, if we return to the broad definition of textons as fundamental micro-structures in natural images, specifically natural textures, then the above mentioned method represents just one way of learning textons. According to [Zhu 05], alternative approaches include ICA [Hyvärinen 09], transformed component analysis (TCA) [Frey 99], and dictionary learning based on sparse coding [Olshausen 97], as well as their own three-level generative image model for learning textons [Zhu 05].

Dictionary learning based on sparse and redundant representation modelling has received a considerable attention in image processing community over the last two decades. These methods assume that a signal, i.e., an image, can be described as a linear combination of few atoms from a dictionary, which is learnt from the data using sparse coding. Therefore, dictionary atoms can be regarded as textons, since they represent perceptual elements of an image. The seminal work on this topic was presented in [Olshausen 97], where the obtained trained atoms were very similar to the receptive fields, which were until then described by Gabor filters (see also Section A.2). Other dictionary learning methods include the method of optimal directions [Engan 99], generalized principal component analysis (PCA) [Vidal 05], K-SVD [Aharon 06], multi-scale KSVD [Mairal 08], etc. Dictionary can also be built based on the mathematical model, where atoms represent some basis functions that have analytic formulation. They have the advantage of being mathematically sound and they allow fast implicit implementation. On the other hand, trained dictionaries enable greater flexibility and the ability to adapt to the specific data (see [Rubinstein 10] for an excellent review of dictionary-based methods). Recently, they have been also used for texture modelling in, e.g., [Peyré 07].

A.5.2 Textons and texton histograms: original definition

2D textons from [Malik 99, Malik 01], as prototypes of filter responses, are obtained by K-means clustering of outputs of linear oriented Gaussian derivative filters at multiple scales. Let us assume for know a more general filtering approach, where the filter bank is not specified upfront, i.e., it consists of any kind of oriented multi-scale filters. Furthermore, let $f_n(p)$ denote the filter response to the n^{th} filter G_n from the filter bank at pixel p.[2] Prior to computing textons,

[2]Note that in Sections A.2 and A.3, G_n and $f_n(p)$ were used to denote the n^{th} Gabor filter and its response, respectively, while here we start from a more general approach.

a contrast normalization step is suggested in [Malik 01], which normalizes the filter outputs in the following way:

$$f_n(p) = f_n(p)\frac{\log\left(1+\frac{\|\mathbf{f}(p)\|}{0.03}\right)}{\|\mathbf{f}(p)\|}, \quad n = 1,\ldots,N_f, \tag{A.6}$$

where $\|\mathbf{f}(p)\|$ is the L_2 norm of the filter responses at the pixel p, and N_f is the total number of filters in the filter bank. Then textons are obtained by first clustering normalized high-dimensional vectors of filter outputs with the K-means algorithm [Duda 73]. The criterion for this algorithm is to find K centres, such that after assigning each data vector to the nearest center, the sum of squared distances from the center is minimized. *Textons* are then defined as the K cluster centres, each being a vector of dimensionality N_f. Each pixel p is assigned to one of the K textons. Let $T(p)$ denote this pixel-to-texton mapping, which takes one of the K possible values, for K textons, i.e., $T(p) = n$, $n = 1,\ldots,K$.

In [Malik 01], the normalized texton histogram in the area surrounding each pixel was used to estimate texturedeness at that pixel, while in, e.g., [Leung 99], this histogram was computed over the whole texture and was used for its representation. Therefore, for some block $B + l$ (centred at the position l), its normalized texton histogram $h^{(l)}$ is defined as

$$h^{(l)}(n) = \frac{1}{\#(B+l)}\sum_{p\in(B+l)}\xi[T(p)=n], \quad n = 1,\ldots,K. \tag{A.7}$$

This texton histogram has K bins (K is the number of textons), where each bin n contains the number of pixels in $B + l$ that are mapped to the texton n [Malik 01]. In the equation above, T is pixel-to-texton mapping defined earlier and ξ is the indicator function, i.e., it returns one if its argument is true and zero otherwise.

Textons and texton histograms can be very useful for texture analysis, as previously shown in [Leung 01, Malik 01, Cula 04, Martin 04, Varma 05, Arbelaez 11] for image segmentation and texture classification. As mentioned above, they were originally computed using even and odd symmetric Gaussian derivatives as filters in [Malik 01, Leung 01], but the above described analysis can be equivalently performed for any choice of filter bank, thus different filter banks have been proposed over the years (see [Varma 05] for a short overview).

B

Publications

B.1 Publications in international journals

1. T. Ružić, A. Pižurica and W. Philips. *Neighbourhood-consensus message passing as a framework for generalized iterated conditional expectations.* Pattern Recognition Letters, vol. 33, pages 309-318, February 2012.

2. B. Cornelis, T. Ružić, E. Gezels, A. Dooms, A. Pižurica, L. Platiša, J. Cornelis, M. Martens, M. De Mey and I. Daubechies. *Crack detection and inpainting for virtual restoration of paintings: The case of the Ghent Altarpiece.* Signal Processing, vol. 93, no. 3, pages 605-619, March 2013.

3. T. Ružić and A. Pižurica. *Context-aware patch-based image inpainting using Markov random field modelling.* IEEE Trans. on Image Proc. (submitted).

B.2 Book chapters

1. A. Pižurica, L. Platiša, T. Ružić, B. Cornelis, A. Dooms, M. Martens, M. De Mey and I. Daubechies. *Virtual restoration and mathematical analysis of pearls in the Adoration of the Mystic Lamb.* In D. Praet and M. Martens, editeurs, Het Lam Gods Series of Lectures (to appear). 2013.

2. L. Platiša, B. Cornelis, T. Ružić, A. Pižurica, A. Dooms, M. Martens, M. De Mey and I. Daubechies. *Spatiogram features to characterize pearls and beads and other small ball-shaped objects in paintings.* In M. De Mey, M. Martens and C. Stroo, editeurs, Vision and material: interaction between art and science in Jan Van Eyck's time. vol. 6, pages 315-329, KVAB Press. 2012.

B.3 Publications in international and national conferences (P1, C1)

1. B. Goossens, T. Ružić, H. Q. Luong and A. Pižurica. *Adaptive non-local means filtering of images corrupted by colored noise.* In UGent-FirW Doctoraatssymposium, 9e, pages 136-137, 2008.

2. T. Ružić, A. Pižurica and W. Philips. *Efficient inference engine for Ising Markov random field model.* In In Proceedings of Annual Workshop on Circuits, Systems and Signal Processing (ProRISC), 2009.

3. H. Q. Luong, T. Ružić, A. Pižurica and W. Philips. *Single image super-resolution using sparsity constraints and non-local similarities at multiple resolution scales.* In Proceedings of SPIE, volume 7723, 2010.

4. T. Ružić, A. Pižurica and W. Philips. *Neighbourhood-consensus message passing and its potentials in image processing applications.* In J. T. Astola and K. O. Egiazarian, editeurs, Image Processing: Algorithms and Systems IX; Proceedings of SPIE, volume 7870, 2011.

5. T. Ružić, B. Cornelis, L. Platiša, A. Pižurica, A. Dooms, W. Philips, M. Martens, M. De Mey and I. Daubechies. *Virtual restoration of the Ghent Altarpiece using crack detection and inpainting.* In Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS), pages 417-428, 2011.

6. T. Ružić, H. Q. Luong, A. Pižurica and W. Philips. *Single image example-based super-resolution using cross-scale patch matching and Markov random field modelling.* In M. Kamel and A. Campilho, editeurs, Proceedings of Int. Conf. on Image Analysis and Recognition (ICIAR), pages 11-20, 2011.

7. L. Platiša, B. Cornelis, T. Ružić, A. Pižurica, A. Dooms, M. Martens, M. De Mey and I. Daubechies. *Spatiogram features to characterize pearls in paintings.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), pages 801-804, 2011.

8. T. Ružić and A. Pižurica. *Texture and color descriptors as a tool for context-aware patch-based image inpainting.* In Image Processing: Algorithms and Systems X; and Parallel Processing for Imaging Applications II; Proceedings of SPIE, volume 8295, 2012.

9. T. Ružić, A. Pižurica and W. Philips. *Markov random field based image inpainting with context-aware label selection.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), pages 1733-1736, 2012.

10. T. Ružić, A. Pižurica and W. Philips. *Exploring contour and texture features for context-aware patch-based inpainting.* In Proceedings

of Symp. on Signal Processing, Image Processing and Artificial Vision (STSIVA), pages 1-5, 2013.

11. T. Ružić and A. Pižurica. *Context-aware image inpainting with application to virtual restoration of old paintings.* In IEICE Information and Communication Technology Forum (ICTF), pages 1-8, 2013.

B.4 Abstracts in international and national conferences

1. T. Ružić, B. Cornelis, L. Platiša, A. Pižurica, A. Dooms, M. Martens, M. De Mey and I. Daubechies. *Craquelure inpainting in artwork.* In Vision and material: interaction between art and science in Jan van Eyck's time, Abstracts, 2010.

2. L. Platiša, B. Cornelis, T. Ružić, A. Pižurica, A. Dooms, M. Martens, M. De Mey and I. Daubechies. *Pearls and beads in Jan van Eyck's paintings.* In Vision and material: Interaction between art and science in Jan van Eyck's time, Abstracts, 2010.

3. B. Cornelis, L. Platiša, T. Ružić, A. Dooms, A. Pižurica, M. Martens, M. De Mey and I. Daubechies. *Teaching a computer about shapes in paintings.* In Vision and material: Interaction between art and science in Jan van Eyck's time, Abstracts, 2010.

4. T. Ružić. *Single-image example-based super-resolution using Markov random fields.* In UGent-FirW Doctoraatssymposium, 11e, 2010.

5. B. Cornelis, T. Ružić, E. Gezels, A. Dooms, A. Pižurica, L. Platiša, M. Martens, P. Schelkens, M. De Mey and I. Daubechies. *Crack detection and inpainting for virtual restoration of paintings: The case of the Ghent Altarpiece.* In International Workshop on Image Processing for Art Investigation (IP4AI), Abstracts, 2011.

6. L. Platiša, B. Cornelis, T. Ružić, A. Pižurica, A. Dooms, P. Schelkens, M. Martens, M. De Mey and I. Daubechies. *Spatiogram features to characterize pearls in paintings.* In International Workshop on Image Processing for Art Investigation (IP4AI), Abstracts, 2011.

7. L. Platiša, B. Cornelis, T. Ružić, A. Pižurica, A. Dooms, M. Martens, M. De Mey and I. Daubechies. *Spatiogram features to characterize pearls in the Ghent Altarpiece.* In Symposium XVIII for the Study of Underdrawing and Technology in Painting, Abstracts, 2012.

Bibliography

[Abas 03] F. S. Abas & L. Martinez. *Classification of painting cracks for content-based analysis.* In Machine Vision Applications in Industrial Inspection XI; Proceedings of SPIE, 2003.

[Abas 04] F. S. Abas. *Analysis of Craquelure Patterns for Content-based Retrieval.* PhD thesis, University of Southampton, UK, 2004.

[Agarwala 04] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin & M. Cohen. *Interactive digital photomontage.* ACM Trans. Graph., vol. 23, no. 3, pages 294–302, August 2004.

[Aharon 06] M. Aharon, M. Elad & A. Bruckstein. *K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation.* IEEE Trans. on Sig. Proc., vol. 54, no. 11, pages 4311–4322, November 2006.

[Alam 00] M. S. Alam, J. G. Bognar, R. C. Hardie & B. J. Yasuda. *Infrared image registration and high-resolution reconstruction using multiple translationally shifted aliased video frames.* IEEE Trans. on Instrumentation and Measurement, vol. 49, no. 5, pages 915–923, October 2000.

[Allebach 96] J. Allebach & P. W. Wong. *Edge-directed interpolation.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), volume 3, pages 707–710, 1996.

[Aly 03] H. H. Aly & E. Dubois. *Regularized image up-sampling using a new observation model and the level set method.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), pages 665–668, 2003.

[Aly 05] H. H. Aly & E. Dubois. *Image up-sampling using total-variation regularization with a new observation model.* IEEE Trans. on Image Proc., vol. 14, no. 10, pages 1647–1659, October 2005.

[Andrey 98] P. Andrey & P. Tarroux. *Unsupervised segmentation of Markov Random Field modeled textured images using selectionist relaxation.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 20, no. 3, pages 252–262, March 1998.

[Anupam 10] Anupam, P. Goyal & S. Diwakar. *Fast and enhanced algorithm for exemplar based image inpainting.* In Proceedings of Pacific-Rim Symposium on Image and Video Technology (PSIVT), pages 325–330, 2010.

[Arbelaez 11] P. Arbelaez, M. Maire, C. Fowlkes & J. Malik. *Contour detection and hierarchical image segmentation.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 33, no. 5, pages 898–916, May 2011.

[Arya 93] S. Arya & D. M. Mount. *Approximate nearest neighbor queries in fixed dimensions.* In Proceedings of ACM-SIAM Symposium on Discrete algorithms (SODA), pages 271–280, 1993.

[Ashikhmin 01] M. Ashikhmin. *Synthesizing natural textures.* In Proceedings of Symp. on Interactive 3D graphics, pages 217–226, 2001.

[Baker 02] S. Baker & T. Kanade. *Limits on super-resolution and how to break them.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 24, no. 9, pages 1167–1183, September 2002.

[Ballester 01] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro & J. Verdera. *Filling-in by joint interpolation of vector fields and gray levels.* IEEE Trans. on Image Proc., vol. 10, no. 8, pages 1200–1211, August 2001.

[Baraniuk 10] R. Baraniuk, V. Cevher, M. Duarte & C. Hegde. *Model-based compressive sensing.* IEEE Trans. on Information Theory, vol. 56, pages 1982–2001, April 2010.

[Barnard 89] S. Barnard. *Stochastic stereo matching over scale.* Int. Journal of Computer Vision, vol. 3, no. 1, pages 17–32, May 1989.

[Barnes 09] C. Barnes, E. Shechtman, A. Finkelstein & D. B. Goldman. *PatchMatch: a randomized correspondence algorithm for structural image editing.* In ACM SIGGRAPH, pages 24:1–24:11, 2009.

[Barni 00] M. Barni, F. Bartolini & V. Cappellini. *Image processing for virtual restoration of artworks.* IEEE Multimedia, vol. 7, no. 2, pages 34–37, April-June 2000.

[Barnsley 88] M. Barnsley. Fractals Everywhere. Academic Press Professional, Inc., 1988.

[Battiato 02] S. Battiato, G. Gallo & F. Stanco. *A locally adaptive zooming algorithm for digital images.* Image and Vision Computing, vol. 20, pages 805–812, September 2002.

[Bellman 62] R. E. Bellman & S. E. Dreyfus. Applied Dynamic Programming. Princeton University Press, 1962.

[Berrou 93] C. Berrou, A. Glavieux & P. Thitimajshima. *Near Shannon limit error-correcting coding and decoding: Turbo codes.* In Proceedings of IEEE Int. Conf. on Communications, pages 1064–1070, 1993.

[Bertalmio 00] M. Bertalmio, G. Sapiro, V. Caselles & C. Ballester. *Image inpainting.* In SIGGRAPH, pages 417–424, 2000.

[Bertalmio 01] M. Bertalmio, A. L. Bertozzi & G. Sapiro. *Navier-Stokes, fluid dynamics, and image and video inpainting.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 355–362, 2001.

[Bertalmio 03] M. Bertalmio, L. A. Vese, G. Sapiro & S. Osher. *Simultaneous structure and texture image inpainting.* IEEE Trans. on Image Proc., vol. 12, no. 8, pages 882–889, August 2003.

[Bertalmio 06] M. Bertalmio. *Strong-continuation, contrast-invariant inpainting with a third-order optimal PDE.* IEEE Trans. on Image Proc., vol. 15, no. 7, pages 1934–1938, July 2006.

[Besag 74] J. Besag. *Spatial interaction and the statistical analysis of lattice systems.* Journal of the Royal Statistical Society. Series B, vol. 36, no. 2, pages 192–236, 1974.

[Besag 86] J. Besag. *On the statistical analysis of dirty pictures (with discussion).* Journal of the Royal Statistical Society. Series B, vol. 48, no. 3, pages 259–302, May 1986.

[Bilbro 88] G. Bilbro, R. Mann, T. K. Miller, W. E. Synder, D. E. Van den Bout & M. White. *Simulated annealing using the mean field approximation.* In IEEE Conference on Neural Information Processing Systems, 1988.

[Bishop 06] C. M. Bishop. Pattern Recognition and Machine Learning. Springer-Verlag New York, Inc., 2006.

[Blake 87] A. Blake & A. Zisserman. Visual Reconstruction. The MIT Press, Cambridge, Massachusetts, 1987.

[Blake 11] A. Blake, P. Kohli & C. Rother, editeurs. Markov Random Fields for Vision and Image Processing. The MIT Press, Cambridge, Massachusetts, 2011.

[Bleyer 11] M. Bleyer, C. Rother, P. Kohli, D. Scharstein & S. N. Sinha. *Object stereo - Joint stereo matching and object segmentation.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 3081–3088, 2011.

[Borman 99] S. Borman & R. L. Stevenson. *Super-resolution from image sequences - a review.* In Proceedings of the Midwest Symposium on Systems and Circuits, pages 374–378, 1999.

[Borman 04] S. Borman. *Topics in Multiframe Superresolution Restoration.* PhD thesis, University of Notre Dame, IN, 2004.

[Bose 01] N. K. Bose, S. Lertrattanapanich & J. Koo. *Advances in superresolution using L-curve.* In Proceedings of IEEE Int. Symp. on Circuits and Systems, volume 2, pages 433–436, 2001.

[Bovik 90] A. C. Bovik, M. Clark & W. S. Geisler. *Multichannel texture analysis using localized spatial filters.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 12, no. 1, pages 55–73, January 1990.

[Boykov 01a] Y. Boykov & M.-P. Jolly. *Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images.* In Proceedings of Int. Conf. on Computer Vision (ICCV), pages 105–112, 2001.

[Boykov 01b] Y. Boykov, O. Veksler & R. Zabih. *Fast approximate energy minimization via graph cuts.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 23, no. 11, pages 1222–1239, November 2001.

[Boykov 04] Y. Boykov & V. Kolmogorov. *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 26, no. 9, pages 1124–1137, September 2004.

[Buades 05] A. Buades, B. Coll & J. M. Morel. *A review of image denoising algorithms, with a new one.* Multiscale Modeling and Simulation, vol. 4, no. 2, pages 490–530, 2005.

[Buades 08] A. Buades, B. Coll & J.-M. Morel. *Nonlocal image and movie denoising.* Int. Journal of Computer Vision, vol. 76, no. 2, pages 123–139, February 2008.

[Bucklow 97] S. Bucklow. *The description of craquelure patterns.* Studies in Conservation, vol. 42, 1997.

[Bucklow 98] S. Bucklow. *A stylometric analysis of craquelure.* Computers and Humanities, vol. 31, 1998.

[Bugeau 10] A. Bugeau, M. Bertalmio, V. Caselles & G. Sapiro. *A comprehensive framework for image inpainting.* IEEE Trans. on Image Proc., vol. 19, no. 10, pages 2634–2645, October 2010.

[Campbell 68] F. W. Campbell & J. G. Robson. *Application of Fourier analysis to the visibility of gratings.* Journal of Physiology, vol. 197, no. 3, pages 551–566, August 1968.

[Canny 86] J. Canny. *A computational approach to edge detection.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 8, no. 6, pages 679–698, November 1986.

[Carey 99] W. K. Carey, D. B. Chuang & S. S. Hemami. *Regularity-preserving image interpolation.* IEEE Trans. on Image Proc., vol. 8, no. 9, pages 1293–1297, September 1999.

[Cerny 85] V. Cerny. *Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm.* Journal of Optimization Theory and Applications, vol. 45, no. 1, pages 41–51, January 1985.

[Cevher 10] V. Cevher, P. Indyk, L. Carin & R. G. Baraniuk. *Sparse signal recovery and acquisition with graphical models.* IEEE Sig. Proc. Magazine, vol. 27, no. 6, pages 92–103, November 2010.

[Chan 01a] T. F. Chan & J. Shen. *Morphologically invariant PDE inpaintings.* Rapport technique, Univ. California, Los Angeles, 2001.

[Chan 01b] T. F. Chan & J. Shen. *Non-Texture inpainting by curvature-driven diffusions (CDD).* Journal of Visual Communication and Image Representation, vol. 12, pages 436–449, 2001.

[Chang 95] S. G. Chang, Z. Cvetkovic & M. Vetterli. *Resolution enhancement of images using wavelet transform extrema interpolation.* In Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pages 2379–2382, 1995.

[Chang 04] H. Chang, D.-Y. Yeung & Y. Xiong. *Super-resolution through neighbor embedding.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 275–282, 2004.

[Chellappa 85] R. Chellappa & S. Chatterjee. *Classification of textures using Gaussian Markov random fields.* IEEE Trans. on Acoustic, Speech, and Signal Processing, vol. 33, no. 4, pages 959–963, August 1985.

[Chen 89] C. C. Chen, J. S. Daponte & M. D. Fox. *Fractal feature analysis and classification in medical imaging.* IEEE Trans. on Medical Imaging, vol. 8, pages 133–142, June 1989.

[Chen 99] C.-C. Chen & C.-C. Chen. *Filtering methods for texture discrimination.* Pattern Recognition Letters, vol. 20, no. 8, pages 783–790, August 1999.

[Cheng 05] W.-H. Cheng, C.-W. Hsieh, S.-K. Lin, C.-W. Wang & J.-L. Wu. *Robust algorithm for exemplar-based image inpainting.* In Int. Conf. of Computer Graphics, Imaging and Vision (CGIV), pages 64–69, 2005.

[Chou 90] P. B. Chou & C. M. Brown. *The theory and practice of Bayesian image labeling.* Int. Journal of Computer Vision, vol. 4, no. 3, pages 185–210, June 1990.

[Clausi 00] D. A. Clausi & M. E. Jernigan. *Designing Gabor filters for optimal texture separability.* Pattern Recognition, vol. 33, no. 11, pages 1835–1849, November 2000.

[Conners 83] R. W. Conners, C. W. McMillin, K. Lin & R. E. Vasquez-Espinosa. *Identifying and locating surface defects in wood: part of an automated lumber processing system.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 5, no. 6, pages 573–583, June 1983.

[Cornelis 11] B. Cornelis, T. Ružić, E. Gezels, A. Dooms, A. Pižurica, L. Platiša, M. Martens, P. Schelkens, M. De Mey & I. Daubechies. *Crack detection and inpainting for virtual restoration of paintings: The case of the Ghent Altarpiece.* In International Workshop on Image Processing for Art Investigation (IP4AI), Abstracts, 2011.

[Cornelis 13] B. Cornelis, T. Ružić, E. Gezels, A. Dooms, A. Pižurica, L. Platiša, J. Cornelis, M. Martens, M. De Mey & I. Daubechies. *Crack detection and inpainting for virtual restoration of paintings: The case of the Ghent Altarpiece.*

Signal Processing, vol. 93, no. 3, pages 605–619, March 2013.

[Criminisi 04] A. Criminisi, P. Perez & K.Toyama. *Region filling and object removal by exemplar-based image inpainting.* IEEE Trans. on Image Proc., vol. 13, no. 9, pages 1200–1212, September 2004.

[Cross 83] G. C. Cross & A. K. Jain. *Markov random field texture models.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 5, no. 1, pages 25–39, January 1983.

[Cula 01] O. G. Cula & K. J. Dana. *Compact representation of bidirectional texture functions.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 1041–1047, 2001.

[Cula 04] O. G. Cula & K. J. Dana. *3D Texture Recognition Using Bidirectional Feature Histograms.* Int. Journal of Computer Vision, vol. 59, no. 1, pages 33–60, August 2004.

[Dabov 07] K. Dabov, A. Foi, V. Katkovnik & K. O. Egiazarian. *Image denoising by sparse 3-D transform-domain collaborative filtering.* IEEE Trans. on Image Proc., vol. 16, no. 8, pages 2080–2095, August 2007.

[Dai 09] S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong & A. K. Katsaggelos. *SoftCuts: a soft edge smoothness prior for color image super-resolution.* IEEE Trans. on Image Proc., vol. 18, no. 5, pages 969–981, May 2009.

[Datsenko 07] D. Datsenko & M. Elad. *Example-based single document image super-resolution: A global MAP approach with outlier rejection.* Multidimensional Syst. Signal Process., vol. 18, pages 103–121, September 2007.

[Daubechies 92] I. Daubechies. Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[Daubechies 12] I. Daubechies. *Developing mathematical tools to investigate art.* Bridges 2012 Proceedings, 2012.

[Daugman 85] J. G. Daugman. *Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters.* Journal of the Optical Society of America A (JOSA A), vol. 2, pages 1160–1169, July 1985.

[Devalois 82] R. L. Devalois, D. G. Albrecht & L.G. Thorell. *Spatial-frequency selectivity of cells in macaque visual cortex.* Vision Research, vol. 22, no. 5, pages 545–559, 1982.

[Di Zenzo 86] S. Di Zenzo. *A note on the gradient of a multi-image.* Computer Vision, Graphics, and Image Processing, vol. 33, no. 1, pages 116–125, January 1986.

[Dinic 70] E. A. Dinic. *Algorithm for solution of a problem of maximum flow in networks with power estimation.* Doklady Akademii Nauk SSSR, vol. 194, no. 4, pages 1277–1280, 1970.

[Drori 03] I. Drori, D. Cohen-Or & H. Yeshurun. *Fragment-based image completion.* In SIGGRAPH, pages 303–312, 2003.

[Duda 73] R. O. Duda & P. E. Hart. Pattern Classification and Scene Analysis. John Willey & Sons, New York, 1973.

[Ebrahimi 07] M. Ebrahimi & E. Vrscay. *Solving the inverse problem of image zooming using "self-examples".* In Proceedings of Int. Conf. on Image Analysis and Recognition (ICIAR), pages 117–130, 2007.

[Efros 99] A. A. Efros & T. K. Leung. *Texture synthesis by non-parametric sampling.* In Proceedings of Int. Conf. on Computer Vision (ICCV), pages 1033–1038, 1999.

[Efros 01] A. A. Efros & W. T. Freeman. *Image quilting for texture synthesis and transfer.* In SIGGRAPH, pages 341–346, 2001.

[Elad 97] M. Elad & A. Feuer. *Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images.* IEEE Trans. on Image Proc., vol. 6, no. 12, pages 1646–1658, December 1997.

[Elad 99a] M. Elad & A. Feuer. *Super-resolution reconstruction of image sequences.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 21, no. 9, pages 817–834, September 1999.

[Elad 99b] M. Elad & A. Feuer. *Superresolution restoration of an image sequence: adaptive filtering approach.* IEEE Trans. on Image Proc., vol. 8, no. 3, pages 387–395, March 1999.

[Elad 05] M. Elad, J. L. Starck, P. Querre & D. L. Donoho. *Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA).* Applied and Computational Harmonic Analysis, vol. 19, no. 3, pages 340–358, November 2005.

[Elad 09] M Elad & D. Datsenko. *Example-based regularization deployed to super-resolution reconstruction of a single image*. Computer Journal, vol. 521, no. 1, pages 15–30, January 2009.

[Engan 99] K. Engan, S. O. Aase & J. Hakon Husoy. *Method of optimal directions for frame design*. In Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), volume 5, pages 2443–2446, 1999.

[Eren 97] P. E. Eren, M. I. Sezan & A. M. Tekalp. *Robust, object-based high-resolution image reconstruction from low-resolution video*. IEEE Trans. on Image Proc., vol. 6, no. 10, pages 1446–1451, October 1997.

[Esedoglu 02] S. Esedoglu & J. Shen. *Digital inpainting based on the Mumford-Shah-Euler image model*. European Journal of Applied Mathematics, vol. 13, pages 353–370, August 2002.

[Fadili 09] M.-J. Fadili, J.-L. Starck & F. Murtagh. *Inpainting and zooming using sparse representations*. Computer Journal, vol. 52, no. 1, pages 64–79, January 2009.

[Fang 09] C.-W. Fang & J.-J. J. Lien. *Rapid image completion system using multiresolution patch-based directional and nondirectional approaches*. IEEE Trans. on Image Proc., vol. 18, pages 2769–2779, December 2009.

[Farsiu 04] S. Farsiu, M. D. Robinson, M. Elad & P. Milanfar. *Fast and robust multiframe super resolution*. IEEE Trans. on Image Proc., vol. 13, no. 10, pages 1327–1344, October 2004.

[Fattal 07] R. Fattal. *Image upsampling via imposed edge statistics*. ACM Trans. Graph., vol. 26, no. 3, July 2007.

[Felzenszwalb 04] P. F. Felzenszwalb & D. P. Huttenlocher. *Efficient belief propagation for early vision*. In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 261–268, 2004.

[Ferreira 94] P. J. S. G. Ferreira. *Interpolation and the discrete Papoulis-Gerchberg algorithm*. IEEE Trans. on Sig. Proc., vol. 42, no. 10, pages 2596–2606, October 1994.

[Ford 62] L. Ford & D. Fulkerson. Flows in Networks. Princeton University Press, 1962.

[Franke 82] R. Franke. *Scattered data interpolation: Tests of some methods.* Mathematics of Computation, vol. 38, no. 157, pages 181–200, 1982.

[Freedman 11] G. Freedman & R. Fattal. *Image and video upscaling from local self-examples.* ACM Trans. Graph., vol. 30, pages 12:1–12:11, April 2011.

[Freeman 00] W. T. Freeman, E. C. Pasztor & O. T. Carmichael. *Learning low-level vision.* Int. Journal of Computer Vision, vol. 40, no. 1, pages 24–47, October 2000.

[Freeman 02] W. T. Freeman, T. R. Jones & E. C. Pasztor. *Example-based super-resolution.* IEEE Computer Graphics and Applications, vol. 22, no. 2, pages 56–65, March 2002.

[Frey 98] B. J. Frey & D. J. C. MacKay. *A revolution: Belief propagation in graphs with cycles.* In Neural Information Processing Systems, pages 479–485, 1998.

[Frey 99] B. J. Frey & N. Jojic. *Transformed component analysis: Joint estimation of spatial transformations and image components.* In Proceedings of Int. Conf. on Computer Vision (ICCV), volume 2, pages 1190–1196, 1999.

[Gabor 46] D. Gabor. *Theory of communication.* Journal of the Institution of Electrical Engineers - Part III: Radio and Communication Engineering, vol. 93, no. 26, pages 429–457, November 1946.

[Geman 84] S. Geman & D. Geman. *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 6, pages 721–741, November 1984.

[Geman 86] S. Geman & C. Graffigne. *Markov random field image models and their application to computer vision.* In Proceedings of Int. Cong. Mathematicians, pages 1496–1517, 1986.

[Gerchberg 74] R. W. Gerchberg. *Super-resolution through error energy reduction.* Journal of Modern Optics, vol. 21, no. 9, pages 709–720, 1974.

[Giakoumis 98] I. Giakoumis & I. Pitas. *Digital restoration of painting cracks.* In Proceedings of IEEE Int. Symp. on Circuits and Signals (ISCAS), volume 4, pages 269–272, 1998.

[Giakoumis 06] I. Giakoumis, N. Nikolaidis & I. Pitas. *Digital image processing techniques for the detection and removal of cracks in digitized paintings.* IEEE Trans. on Image Proc., vol. 15, no. 1, pages 178–188, January 2006.

[Gionis 99] A. Gionis, P. Indyk & R. Motwani. *Similarity search in high dimensions via hashing.* In Proceedings of Int. Conf. on Very Large Data Bases (VLDB), pages 518–529, 1999.

[Glasner 09] D. Glasner, S. Bagon & M. Irani. *Super-resolution from a single image.* In Proceedings of Int. Conf. on Computer Vision (ICCV), pages 349–356, 2009.

[Goossens 08] B. Goossens, H. Q. Luong, A. Pižurica & W. Philips. *An improved non-local means algorithm for image denoising.* In International Workshop on Local and Non-Local Approximation in Image Processing (LNLA), 2008.

[Greig 89] D. M. Greig, B. T. Porteous & A. H. Seheult. *Exact maximum a posteriori estimation for binary images.* Journal of the Royal Statistical Society. Series B (Methodological), vol. 51, no. 2, pages 271–279, 1989.

[Guleryuz 06a] O. G. Guleryuz. *Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part I: theory.* IEEE Trans. on Image Proc., vol. 15, no. 3, pages 539–554, March 2006.

[Guleryuz 06b] O. G. Guleryuz. *Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part II: adaptive algorithms.* IEEE Trans. on Image Proc., vol. 15, no. 3, pages 555–571, March 2006.

[HaCohen 10] Y. HaCohen, R. Fattal & D. Lischinski. *Image upsampling via texture hallucination.* In Proceedings of IEEE Int. Conf. on Computational Photography (ICCP), 2010.

[Hanbury 03] A. Hanbury, P. Kammerer & E. Zolda. *Painting crack elimination using viscous morphological reconstruction.* In Proceedings of Int. Conf. on Image Analysis and Processing (ICIAP), pages 226–231, 2003.

[Haralick 73] R. M. Haralick, K. Shanmugam & I. Dinstein. *Textural features for image classification.* IEEE Trans. on Systems, Man and Cybernetics, vol. 3, no. 6, pages 610–621, November 1973.

[Hardie 97] R. C. Hardie, K. J. Barnard & E. E. Armstrong. *Joint MAP registration and high-resolution image estimation*

using a sequence of undersampled images. IEEE Trans. on Image Proc., vol. 6, no. 12, pages 1621–1633, December 1997.

[Hardie 98] R. C. Hardie, K. J. Barnard, J. G. Bognar, E. E. Armstrong & E. A. Watson. *High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system.* Optical Engineering, vol. 37, no. 1, pages 247–260, January 1998.

[Hays 08] J. Hays & A. A. Efros. *Scene completion using millions of photographs.* Commun. ACM, vol. 51, no. 10, pages 87–94, October 2008.

[He 09] L. He & L. Carin. *Exploiting structure in wavelet-based Bayesian compressive sensing.* IEEE Trans. on Sig. Proc., vol. 57, pages 3488–3497, September 2009.

[He 12] K. He & J. Sun. *Statistics of patch offsets for image completion.* In Proceedings of European Conf. on Computer Vision (ECCV), pages 16–29, 2012.

[Hertzmann 01] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless & D. H. Salesin. *Image analogies.* In SIGGRAPH, pages 327–340, 2001.

[Hong 97] M.-C. Hong, M. G. Kang & A. K. Katsaggelos. *An iterative weighted regularized algorithm for improving the resolution of video sequences.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), volume 2, pages 474–477, 1997.

[Huang 07] T. Huang, S. Chen, J. Liu & X. Tang. *Image inpainting by global structure and texture propagation.* In Proceedings of ACM Int. Conf. on Multimedia, pages 517–520, 2007.

[Huang 09] J. Huang, T. Zhang & D. Metaxas. *Learning with structured sparsity.* In Proceedings of Int. Conf. on Machine Learning (ICML), pages 417–424, 2009.

[Hubel 62] D. H. Hubel & T. N. Wiesel. *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex.* Journal of Physiology, vol. 160, pages 106–154, January 1962.

[Hyvärinen 09] A. Hyvärinen, J. Hurri & P. O. Hoyer. Natural Image Statistics: A Probabilistic Approach to Early Computational Vision. Springer-Verlag, New York, 2009.

[Irani 91] M. Irani & S. Peleg. *Improving resolution by image registration.* CVGIP: Graphical Models and Image Processing, vol. 53, no. 3, pages 231–239, May 1991.

[Irani 93] M. Irani & S. Peleg. *Motion analysis for image enhancement: resolution, occlusion, and transparency.* Journal of Visual Communication and Image Representation, vol. 4, pages 324–335, December 1993.

[Jacquin 92] A. E. Jacquin. *Image coding based on a fractal theory of iterated contractive image transformations.* IEEE Trans. on Image Proc., vol. 1, no. 1, pages 18–30, January 1992.

[Jain 89] A. K. Jain. Fundamentals of Digital Image Processing. Prentice-Hall, Englewood Cloffs, NJ, 1989.

[Jain 90] A. K. Jain, F. Farrokhnia & D. H. Alman. *Texture analysis of automotive finishes.* In Proceedings of SME Machine Vision Applications Conference, pages 1–16, 1990.

[Jain 91] A. K. Jain & F. Farrokhnia. *Unsupervised texture segmentation using Gabor filters.* Pattern Recognition, vol. 24, no. 12, pages 1167–1186, December 1991.

[Jain 97] A. K. Jain, N. K. Ratha & S. Lakshmanan. *Object detection using Gabor filters.* Pattern Recognition, vol. 30, no. 2, pages 295–309, February 1997.

[Jensen 95] K. Jensen & D. Anastassiou. *Sub-pixel edge localization and the interpolation of still images.* IEEE Trans. on Image Proc., vol. 4, no. 3, pages 285–295, March 1995.

[Jia 03] J. Jia & C.-K. Tang. *Image repairing: Robust image synthesis by adaptive ND tensor voting.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 643–650, 2003.

[Jiang 02] H. Jiang & C. Moloney. *A new direction adaptive scheme for image interpolation.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), volume 3, pages 369–372, 2002.

[Johnson 08] C. R. Johnson, E. Hendriks, I. Berezhnoy, E. Brevdo, S. Hughes, I. Daubechies, J. Li, E. Postma & J. Z. Wang. *Image processing for artist identification - Computerized analysis of Vincent van Gogh's painting brushstrokes.* IEEE Sig. Proc. Magazine, pages 37–48, July 2008.

[Jolliffe 02] I. T. Jolliffe. Principal Component Analysis. Springer, 2nd edition, 2002.

[Jones 92] D. G. Jones & J. Malik. *Computational framework for determining stereo correspondence from a set of linear spatial filters.* Image and Vision Computing, vol. 10, no. 10, pages 699–708, 1992.

[Joshi 02] M. V. Joshi & S. Chaudhuri. *Super-resolution imaging: Use of zoom as a cue.* In Proceedings of Indian Conf. Vision, Graphics and Image Processing, pages 439–444, 2002.

[Joyeux 99] L. Joyeux, O. Buisson, B. Besserer & S. Boukir. *Detection and removal of line scratches in motion picture films.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 111–114, 1999.

[Julesz 81] B. Julesz. *Textons, the elements of texture perception, and their interactions.* Nature, vol. 290, no. 5802, pages 91–97, March 1981.

[Kanizsa 79] G. Kanizsa. Organization in Vision: Essays on Gestalt Perception. Praeger, New York, 1979.

[Karni 91] A. Karni & D. Sagi. *Where practice makes perfect in texture discrimination: Evidence for primary visual cortex plasticity.* Proceedings of the National Academy of Sciences of the United States of America, vol. 88, no. 11, pages 4966–4970, June 1991.

[Kawai 08] N. Kawai, T. Sato & N. Yokoya. *Image inpainting considering brightness change and spatial locality of textures and its evaluation.* In Proceedings of Pacific-Rim Symposium on Image and Video Technology (PSIVT), pages 271–282, 2008.

[Kim 90] S. P. Kim, N. K. Bose & H. M. Valenzuela. *Recursive reconstruction of high resolution image from noisy undersampled multiframes.* IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. 38, no. 6, pages 1013–1027, June 1990.

[Kim 93] S. P. Kim & W. Y. Su. *Recursive high-resolution reconstruction of blurred multiframe images.* IEEE Trans. on Image Proc., vol. 2, no. 4, pages 534–539, October 1993.

[Kim 08] K. I. Kim & Y. Kwon. *Example-based learning for single-image super-resolution.* In Proceedings of the DAGM symposium on Pattern Recognition, pages 456–465, 2008.

[Kinebuchi 01] K. Kinebuchi, D. D. Muresan & T. W. Parks. *Image interpolation using wavelet-based hidden Markov trees.* In Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), volume 3, pages 1957–1960, 2001.

[King 97] D. King. The Commissar Vanishes. Metropolitan Books, 1997.

[Kingsbury 01] N. Kingsbury. *Complex wavelets for shift invariant analysis and filtering of signals.* Applied and Computational Harmonic Analysis, vol. 10, no. 3, pages 234–253, May 2001.

[Kirkpatrick 83] S. Kirkpatrick, C. D. Gelatt & M. P. Vecchi. *Optimization by simulated annealing.* Science, vol. 220, pages 671–680, May 1983.

[Kohli 07] P. Kohli, M. P. Kumar & P. H. S. Torr. *P^3 and beyond: Solving energies with higher order cliques.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 1–8, 2007.

[Kohli 09] P. Kohli, L. Ladicky & P. H. S. Torr. *Robust higher order potentials for enforcing label consistency.* Int. Journal of Computer Vision, vol. 82, no. 3, pages 302–324, May 2009.

[Kokaram 95a] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald & P. J. W. Rayner. *Detection of missing data in image sequences.* IEEE Trans. on Image Proc., vol. 4, no. 11, pages 1496–1508, November 1995.

[Kokaram 95b] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald & P. J. W. Rayner. *Interpolation of missing data in image sequences.* IEEE Trans. on Image Proc., vol. 4, no. 11, pages 1509–1519, November 1995.

[Kokaram 95c] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald & P. J. W. Rayner. *Interpolation of missing data in image sequences.* IEEE Trans. on Image Proc., vol. 4, no. 11, pages 1509–1519, November 1995.

[Kolmogorov 04] V. Kolmogorov & R. Zabih. *What energy functions can be minimized via graph cuts?* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 24, no. 2, pages 147–159, February 2004.

[Kolmogorov 06] V. Kolmogorov. *Convergent tree-reweighted message passing for energy minimization.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 28, no. 10, pages 1568–1583, October 2006.

[Komodakis 07] N. Komodakis & G. Tziritas. *Image completion using efficient belief propagation via priority scheduling and dynamic pruning.* IEEE Trans. on Image Proc., vol. 16, no. 11, pages 2649–2661, November 2007.

[Konishi 00] S. Konishi & A. L. Yuille. *Statistical cues for domain specific image segmentation with performance analysis.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 1125–1132, 2000.

[Kwatra 03] V. Kwatra, A. Schödl, I. Essa, G. Turk & A. Bobick. *Graphcut textures: Image and video synthesis using graph cuts.* In SIGGRAPH, pages 277–286, 2003.

[Le Meur 11] O. Le Meur, J. Gautier & C. Guillemot. *Examplar-based inpainting based on local geometry.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), pages 3462–3465, 2011.

[Le Meur 12] O. Le Meur & C. Guillemot. *Super-resolution-based inpainting.* In Proceedings of European Conf. on Computer Vision (ECCV), pages 554–567, 2012.

[Lee 03] E. S. Lee & M. G. Kang. *Regularized adaptive high-resolution image reconstruction considering inaccurate subpixel registration.* IEEE Trans. on Image Proc., vol. 12, no. 7, pages 826–837, July 2003.

[Lehmann 99] T. M. Lehmann, C. Gönner & K. Spitzer. *Survey: Interpolation methods in medical image processing.* IEEE Trans. on Medical Imaging, vol. 18, no. 11, pages 1049–1075, 1999.

[Leung 99] T. Leung & J. Malik. *Recognizing surfaces using three-dimensional textons.* In Proceedings of Int. Conf. on Computer Vision (ICCV), pages 1010–1017, 1999.

[Leung 01] T. Leung & J. Malik. *Representing and recognizing the visual appearance of materials using three-dimensional textons.* Int. Journal of Computer Vision, vol. 43, no. 1, 201.

[Li 90] S. Z. Li. *Invariant surface segmentation through energy minimization with discontinuities.* Int. Journal of Computer Vision, vol. 5, no. 2, pages 161–194, November 1990.

[Li 95] S. Z. Li. Markov Random Field Modeling in Computer Vision. Springer-Verlag, London, UK, 1995.

[Li 01] X. Li & M. T. Orchard. *New edge-directed interpolation.* IEEE Trans. on Image Proc., vol. 10, no. 10, pages 1521–1527, October 2001.

[Li 08] M. Li & T. Q. Nguyen. *Markov random field model-based edge-directed image interpolation.* IEEE Trans. on Image Proc., vol. 17, no. 7, pages 1121–1128, July 2008.

[Li 09] S. Z. Li. Markov Random Field Modeling in Image Analysis. Springer Publishing Company, Incorporated, 3rd edition, 2009.

[Liang 01] L. Liang, C. Liu, Y.-Q. Xu, B. Guo & H.-Y. Shum. *Real-time texture synthesis by patch-based sampling.* ACM Trans. Graph., vol. 20, no. 3, pages 127–150, July 2001.

[Lin 04] Z. Lin & H.-Y. Shum. *Fundamental limits of reconstruction-based superresolution algorithms under local translation.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 26, no. 1, pages 83–97, January 2004.

[Lin 08] Z. Lin, J. He, X. Tang & C.-K. Tang. *Limits of learning-based superresolution algorithms.* Int. Journal of Computer Vision, vol. 80, no. 3, pages 406–420, December 2008.

[Liu 01] X. Liu, Y. Yu & H.-Y. Shum. *Synthesizing bidirectional texture functions for real-world surfaces.* In SIGGRAPH, pages 97–106, 2001.

[López 99] A. M. López, F. Lumbreras, J. Serrat & J. J. Villanueva. *Evaluation of methods for ridge and valley detection.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 21, no. 4, pages 327–335, April 1999.

[Luong 05] H. Q. Luong, P. De Smet & W. Philips. *Image interpolation using constrained contrast enhancement techniques.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), pages 998–1001, 2005.

[Luong 07] H. Q. Luong, B. Goossens & W. Philips. *Image upscaling using global multimodal priors.* In J. Blanc-Talon, W. Philips, D. Popescu & P. Scheunders, editeurs, Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS), volume LNCS 4678, pages 473–484, 2007.

[Luong 09] H. Q. Luong. *Advanced Image and Video Resolution Enhancement Techniques.* PhD thesis, Ghent University, Ghent, Belgium, 2009.

[Luong 10] H. Q. Luong, T. Ružić A. Pižurica & W. Philips. *Single image super-resolution using sparsity constraints and non-local similarities at multiple resolution scales.* In Proceedings of SPIE, volume 7723, 2010.

[Mairal 08] J. Mairal, G. Sapiro & M. Elad. *Learning multiscale sparse representations for image and video restoration.* Multiscale Modeling and Simulation, vol. 7, no. 1, pages 214–241, 2008.

[Malfait 97] M. Malfait & D. Roose. *Wavelet-based image denoising using a Markov random field a priori model.* IEEE Trans. on Image Proc., vol. 6, pages 549–565, April 1997.

[Malik 99] J. Malik, S. Belongie, J. Shi & T. Leung. *Textons, contours and regions: Cue integration in image segmentation.* In Proceedings of Int. Conf. on Computer Vision (ICCV), pages 918–925, 1999.

[Malik 01] J. Malik, S. Belongie, T. Leung & J. Shi. *Contour and texture analysis for image segmentation.* Int. Journal of Computer Vision, vol. 43, no. 1, pages 7–27, June 2001.

[Mallat 92] S. Mallat & S. Zhong. *Characterization of signals from multiscale edges.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 14, no. 7, pages 710–732, July 1992.

[Mallat 98] S. G. Mallat. A Wavelet Tour of Signal Processing. New York: Academic, 1998.

[Mallat 09] S. Mallat. A Wavelet Tour of Signal Processing. Academic Press, 3rd edition, 2009.

[Malyshev 91] V. A. Malyshev & R. A. Minlos. Gibbs Random Fields: Cluster Expansions. Kluwer Academic Publishers, 1991.

[Marr 80] D. Marr & E. Hildreth. *Theory of edge detection.* Proceedings of the Royal Society of London. Series B, Biological Sciences, vol. 207, no. 1167, pages 187–217, February 1980.

[Martin 04] D. R. Martin, C. C. Fowlkes & J. Malik. *Learning to detect natural image boundaries using local brightness, color, and texture cues.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 26, no. 5, pages 530–549, May 2004.

[Masnou 98] S. Masnou & J.-M. Morel. *Level lines based disocclusion.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), volume 3, pages 259–263, 1998.

[Masnou 02] S. Masnou. *Disocclusion: a variational approach using level lines.* IEEE Trans. on Image Proc., vol. 11, no. 2, pages 68–76, February 2002.

[Medioni 00] G. Medioni, M.-S. Lee & C.-K. Tang. A Computational Framework for Segmentation and Grouping. Amsterdam: Elseviers Science, 2000.

[Meijering 01] E. H. W. Meijering, W. J. Niessen & M. A. Viergever. *Quantitative evaluation of convolution-based methods for medical image interpolation.* Medical Image Analysis, vol. 5, no. 2, pages 111–126, June 2001.

[Metropolis 53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller & E. Teller. *Equation of state calculations by fast computing machines.* The Journal of Chemical Physics, vol. 21, no. 6, pages 1087–1092, 1953.

[Meyer 79] F. Meyer. *Cytologie quantitative et morphologie mathématique.* PhD thesis, Ecole des Mines, 1979.

[Meyer 95] Y. Meyer & D. Salinger. Wavelets and Operators. Cambridge University Press, 1995.

[Mohen 06] J. Mohen, M. Menu & B. Mottin. Mona Lisa: Inside the Painting. Harry N. Abrams, New York, NY, 2006.

[Morrone 87] M. C. Morrone & R. A. Owens. *Feature detection from local energy.* Pattern Recognition Letters, vol. 6, no. 5, pages 303–313, December 1987.

[Morse 01] B. S. Morse & D. Schwartzwald. *Image magnification using level-set reconstruction.* In Proceedings of Int. Conf. on Computer Vision (ICCV), pages 333–341, 2001.

[Mumford 89] D. Mumford & J. Shah. *Optimal approximations by piecewise smooth functions and associated variational problems.* Communications on Pure and Applied Mathematics, vol. 42, no. 5, pages 577–685, 1989.

[Mumford 94] D. Mumford. *Elastica and computer vision.* In C. L. Bajaj, editeur, Algebraic Geometry and its Applications, pages 491–506. Springer-Verlag, New York, 1994.

[Muresan 01] D. D. Muresan & T. W. Parks. *Optimal recovery approach to image interpolation.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), volume 3, pages 848–851, 2001.

[Muresan 04] D. D. Muresan & T. W. Parks. *Adaptively quadratic (AQua) image interpolation.* IEEE Trans. on Image Proc., vol. 13, no. 5, pages 690–698, May 2004.

[Muresan 05] D. D. Muresan. *Fast edge directed polynomial interpolation.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), pages 990–993, 2005.

[Murphy 99] K. P. Murphy, Y. Weiss & M. I. Jordan. *Loopy belief propagation for approximate inference: An empirical study.* In Proceedings of Uncertainty in AI, pages 467–475, 1999.

[Nene 97] S. A. Nene & S. K. Nayar. *A simple algorithm for nearest neighbor search in high dimensions.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 19, pages 989–1003, September 1997.

[Ng 02] M. K. Ng & N. K. Bose. *Analysis of displacement errors in high-resolution image reconstruction with multisensors.* IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications, vol. 49, no. 6, pages 806–813, 2002.

[Nguyen 00] N. Nguyen & P. Milanfar. *An efficient wavelet-based algorithm for image superresolution.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), volume 2, pages 351–354, 2000.

[Nitzberg 93] M. Nitzberg, D. Mumford & T. Shiota. Filtering, Segmentation, and Depth. Springer-Verlag New York, Inc., 1993.

[Ojala 96] T. Ojala, M. Pietikäinen & D. Harwood. *A comparative study of texture measures with classification based on featured distributions.* Pattern Recognition, vol. 29, no. 1, pages 51–59, January 1996.

[Oliva 01] A. Oliva & A. Torralba. *Modeling the shape of the scene: a holistic representation of the spatial envelope.* Int. Journal of Computer Vision, vol. 42, no. 3, pages 145–175, May 2001.

[Olshausen 97] B. A. Olshausen & D. J. Field. *Sparse coding with an overcomplete basis set: a strategy employed by V1?* Vision Research, vol. 37, no. 23, pages 3311–3325, December 1997.

[Orjuela 13] S. A. Orjuela. *Texture Analysis for the Evaluation of Appearance Changes in Textile Surfaces*. PhD thesis, Ghent University, Ghent, Belgium, 2013.

[Orlin 07] J. B. Orlin. *A faster strongly polynomial time algorithm for submodular function minimization*. In Proceedings of Int. Conf. on Integer Programming and Combinatorial Optimization, pages 240–251, 2007.

[Osindero 06] S. Osindero, M. Welling & G. E. Hinton. *Topographic product models applied to natural scene statistics*. Neural Computation, vol. 18, no. 2, pages 381–414, February 2006.

[Owen 89] A. Owen. *Image segmentation via iterated conditional expectations*. Rapport technique, Department of Statistics, University of Chicago, 1989.

[Papandreou 08] G. Papandreou, P. Maragos & A. Kokaram. *Image inpainting with a wavelet domain hidden Markov tree model*. In Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pages 773–776, 2008.

[Papoulis 75] A. Papoulis. *A new algorithm in spectral analysis and band limited signal extrapolation*. IEEE Trans. on Circuits and Systems, vol. 22, pages 735–742, September 1975.

[Park 03] S. C. Park, M. K. Park & M. G. Kang. *Super-resolution image reconstruction: A technical overview*. IEEE Sig. Proc. Magazine, vol. 20, no. 3, pages 21–36, May 2003.

[Patti 97] A. J. Patti, M. I. Sezan & A. M. Tekalp. *Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time*. IEEE Trans. on Image Proc., vol. 6, no. 8, pages 1064–1076, August 1997.

[Patti 01] A. J. Patti & Y. Altunbasak. *Artifact reduction for set theoretic super resolution image reconstruction with edge adaptive constraints and higher-order interpolants*. IEEE Trans. on Image Proc., vol. 10, no. 1, pages 179–186, January 2001.

[Pearl 88] J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco, CA, 1988.

[Pei 04] S.-C. Pei, Y.-C. Zeng & C.-H. Chang. *Virtual restoration of ancient Chinese paintings using color contrast enhancement and lacuna texture synthesis.* IEEE Trans. on Image Proc., vol. 13, no. 3, pages 416–429, March 2004.

[Peleg 87] S. Peleg, D. Keren & L. Schweitzer. *Improving image resolution using subpixel motion.* Pattern Recognition Letters, vol. 5, no. 3, pages 223–226, March 1987.

[Perona 90] P. Perona & J. Malik. *Detecting and localizing edges composed of steps, peaks and roofs.* In Proceedings of Int. Conf. on Computer Vision (ICCV), pages 52–57, 1990.

[Peterson 89] C. Peterson & B Soderberg. *A new method for mapping optimization problems onto neural networks.* Int. Journal of Neural Systems, vol. 1, pages 3–22, 1989.

[Peyré 07] G. Peyré. *Non-negative sparse modeling of textures.* In Scale Space and Variational Methods in Computer Vision, volume 4485, pages 628–639, 2007.

[Pickup 03] L. C. Pickup, S. J. Roberts & A. Zisserman. *A sampled texture prior for image super-resolution.* In Neural Information Processing Systems, pages 1587–1594, 2003.

[Pižurica 02a] A. Pižurica. *Image Denoising Using Wavelets and Spatial Context Modelling.* PhD thesis, Ghent University, Ghent, Belgium, 2002.

[Pižurica 02b] A. Pižurica, W. Philips, I. Lemahieu & M. Acheroy. *A joint inter- and intrascale statistical model for wavelet based Bayesian image denoising.* IEEE Trans. on Image Proc., vol. 11, no. 5, pages 545–557, May 2002.

[Pižurica 11] A. Pižurica, J. Aelterman, F. Bai, S. Vanloocke, H. Luong, B. Goossens & W. Philips. *On structured sparsity and selected applications in tomographic imaging.* In Wavelets and sparsity XIV; Proceedings of SPIE, volume 8138, 2011.

[Pižurica 13] A. Pižurica, L. Platiša, T. Ružić, B. Cornelis, A. Dooms, M. Martens, M. De Mey & I. Daubechies. *Virtual restoration and mathematical analysis of pearls in the Adoration of the Mystic Lamb.* In D. Praet & M. Martens, editeurs, Het Lam Gods Series of Lectures (to appear). 2013.

[Platiša 11] L. Platiša, B. Cornelis, T. Ružić, A. Pižurica, A. Dooms, M. Martens, M. De Mey & I. Daubechies. *Spatiogram features to characterize pearls in paintings.* In Proceedings

of IEEE Int. Conf. on Image Processing (ICIP), pages 801–804, 2011.

[Poggi 05] G. Poggi, G. Scarpa & J. Zerubia. *Supervised segmentation of remote sensing images based on a tree-structured MRF model.* IEEE Trans. on Geoscience and Remote Sensing, vol. 43, no. 8, pages 1901–1911, August 2005.

[Poggio 85] T. Poggio & C. Koch. *Ill-posed problems in early vision: From computational theory to analogue networks.* Proc. of the Royal Society of London. Series B, Biological Sciences, vol. 226, no. 1244, pages 303–323, December 1985.

[Poli 97] R Poli & G Valli. *An algorithm for real-time vessel enhancement and detection.* Computer methods and programs in biomedicine, vol. 52, no. 1, pages 1–22, January 1997.

[Polidori 97] E. Polidori & J.-L. Dugelay. *Zooming using iterated function systems.* Fractals, Supplementary Issue, vol. 5, April 1997.

[Portilla 00] J. Portilla & E. P. Simoncelli. *A parametric texture model based on joint statistics of complex wavelet coefficients.* Int. Journal of Computer Vision, vol. 40, no. 1, pages 49–70, October 2000.

[Potts 52] R. B. Potts. *Some generalized order-disorder transformations.* Mathematical Proceedings of the Cambridge Philosophical Society, vol. 48, pages 106–109, 1952.

[Protter 09] M. Protter, M. Elad, H. Takeda & P. Milanfar. *Generalizing the nonlocal-means to super-resolution reconstruction.* IEEE Trans. on Image Proc., vol. 18, no. 1, pages 36–51, January 2009.

[Puzicha 97] J. Puzicha, T. Hofmann & J. Buhmann. *Non-parametric similarity measures for unsupervised texture segmentation and image retrieval.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 267–272, 1997.

[Rabaud 05] V. Rabaud & S. Belongie. *Big little icons.* In Computer Vision Applications for the Visually Impaired (CVAVI), 2005.

[Raj 05] A. Raj & R. Zabih. *A graph cut algorithm for generalized image deconvolution.* In Proceedings of Int. Conf. on Computer Vision (ICCV), pages 1048–1054, 2005.

[Rajan 01a] D. Rajan & S. Chaudhuri. *Generalized interpolation and its application in super-resolution imaging.* Image and Vision Computing, vol. 19, no. 13, pages 957–969, November 2001.

[Rajan 01b] D. Rajan & S. Chaudhuri. *Simultaneous estimation of super-resolved intensity and depth maps from low resolution defocused observations of a scene.* In Proceedings of Int. Conf. on Computer Vision (ICCV), pages 113–118, 2001.

[Rajan 02] D. Rajan & S. Chaudhuri. *An MRF-based approach to generation of super-resolution images from blurred observations.* Journal of Mathematical Imaging and Vision, vol. 16, no. 1, pages 5–15, January 2002.

[Randen 99] T. Randen & J. H. Husøy. *Filtering for texture classification: A comparative study.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 21, no. 4, pages 291–310, April 1999.

[Ratakonda 98] K. Ratakonda & N. Ahuja. *POCS based adaptive image interpolation.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), pages 203–207, 1998.

[Rhee 99] S. H. Rhee & M. G. Kang. *DCT-based regularized algorithm for high-resolution image reconstruction.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), volume 3, pages 184–187, 1999.

[Rignot 90] E. Rignot & R. Kwok. *Extraction of textural features in SAR images: Statistical model and sensitivity.* In Proceedings of Int. Geoscience and Remote Sensing Symp., pages 1979–1982, 1990.

[Rikert 99] T. D. Rikert, M. J. Jones & P. A. Viola. *A cluster-based statistical model for object detection.* In Proceedings of Int. Conf. on Computer Vision (ICCV), pages 1046–1053, 1999.

[Romberg 01] J. K. Romberg, H. Choi & R. G. Baraniuk. *Bayesian tree-structured image modeling using wavelet-domain hidden Markov models.* IEEE Trans. on Image Proc., vol. 10, no. 7, pages 1056–1068, July 2001.

[Rosenfeld 76] A. Rosenfeld, R. Hummer & S. Zucker. *Scene labeling by relaxation operations.* IEEE Trans. on Systems, Man and Cybernetics, vol. 6, no. 6, pages 420–433, June 1976.

[Roth 05] S. Roth & M. J. Black. *Fields of experts: A framework for learning image priors.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 860–867, 2005.

[Rother 04] C. Rother, V. Kolmogorov & A. Blake. *'GrabCut' - interactive foreground extraction using iterated graph cuts.* ACM Trans. Graph., vol. 23, no. 3, pages 309–314, August 2004.

[Rother 07] C. Rother, V. Kolmogorov, V. Lempitsky & M. Szummer. *Optimizing binary MRFs via extended roof duality.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 1–8, 2007.

[Roweis 00] S. T. Roweis & L. K. Saul. *Nonlinear dimensionality reduction by locally linear embedding.* Science, vol. 290, pages 2323–2326, December 2000.

[Rubinstein 10] R. Rubinstein, A. M. Bruckstein & M. Elad. *Dictionaries for sparse representation modeling.* Proceedings of the IEEE, vol. 98, no. 6, pages 1045–1057, June 2010.

[Rudin 92] L. I. Rudin, S. Osher & E. Fatemi. *Nonlinear total variation based noise removal algorithms.* Physics D, vol. 60, no. 1-4, pages 259–268, November 1992.

[Ružić 09] T. Ružić, A. Pižurica & W. Philips. *Efficient inference engine for Ising Markov random field model.* In Proceedings of Annual Workshop on Circuits, Systems and Signal Processing (ProRISC), 2009.

[Ružić 10] T. Ružić, B. Cornelis, L. Platiša, A. Pižurica, A. Dooms, M. Martens, M. De Mey & I. Daubechies. *Craquelure inpainting in artwork.* In Vision and material: Interaction between art and science in Jan van Eyck's time, Abstracts, 2010.

[Ružić 11a] T. Ružić, B. Cornelis, L. Platiša, A. Pižurica, A. Dooms, W. Philips, M. Martens, M. De Mey & I. Daubechies. *Virtual restoration of the Ghent Altarpiece using crack detection and inpainting.* In Proceedings of Advanced Concepts for Intelligent Vision Systems (ACIVS), pages 417–428, 2011.

[Ružić 11b] T. Ružić, H. Q. Luong, A. Pižurica & W. Philips. *Single image example-based super-resolution using cross-scale patch matching and Markov random field modelling.* In M. Kamel & A. Campilho, editeurs, Proceedings of Int.

Conf. on Image Analysis and Recognition (ICIAR), pages 11–20, 2011.

[Ružić 11c] T. Ružić, A. Pižurica & W. Philips. *Neighbourhood-consensus message passing and its potentials in image processing applications.* In J. T. Astola & K. O. Egiazarian, editeurs, Image Processing: Algorithms and Systems IX; Proceedings of SPIE, volume 7870, 2011.

[Ružić 12a] T. Ružić & A. Pižurica. *Texture and color descriptors as a tool for context-aware patch-based image inpainting.* In Image Processing: Algorithms and Systems X; and Parallel Processing for Imaging Applications II; Proceedings of SPIE, volume 8295, 2012.

[Ružić 12b] T. Ružić, A. Pižurica & W. Philips. *Markov random field based image inpainting with context-aware label selection.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), pages 1733–1736, 2012.

[Ružić 12c] T. Ružić, A. Pižurica & W. Philips. *Neighbourhood-consensus message passing as a framework for generalized iterated conditional expectations.* Pattern Recognition Letters, vol. 33, pages 309–318, February 2012.

[Ružić 13a] T. Ružić & A. Pižurica. *Context-aware image inpainting with application to virtual restoration of old paintings.* In IEICE Information and Communication Technology Forum (ICTF), pages 1–8, 2013.

[Ružić 13b] T. Ružić, A. Pižurica & W. Philips. *Context-aware patch-based image inpainting using Markov random field modelling.* IEEE Trans. on Image Proc. (submitted), 2013.

[Ružić 13c] T. Ružić, A. Pižurica & W. Philips. *Exploring contour and texture features for context-aware patchbased inpainting.* In Proceedings of Symp. on Signal Processing, Image Processing and Artificial Vision (STSIVA), pages 1–5, 2013.

[Scarpa 09] G. Scarpa, R. Gaetano, M. Haindl & J. Zerubia. *Hierarchical multiple Markov chain model for unsupervised texture segmentation.* IEEE Trans. on Image Proc., vol. 18, no. 8, pages 1830–1843, August 2009.

[Schroff 06] F. Schroff, A. Criminisi & A. Zisserman. *Single-histogram class models for image segmentation.* In Proceedings of Indian Conf. Vision, Graphics and Image Processing, pages 82–93, 2006.

[Schultz 96] R. R. Schultz & R. L. Stevenson. *Extraction of high-resolution frames from video sequences.* IEEE Trans. on Image Proc., vol. 5, no. 6, pages 996–1011, June 1996.

[Selesnick 05] I. W. Selesnick, R. G. Baraniuk & N. C. Kingsbury. *The dual-tree complex wavelet transform.* IEEE Sig. Proc. Magazine, vol. 22, no. 6, pages 123–151, November 2005.

[Serra 99] J. Serra. *Les treillis visqueux.* Rapport technique, Ecole des Mines de Paris, 1999.

[Shah 99] N. R. Shah & A. Zakhor. *Resolution enhancement of color video sequences.* IEEE Trans. on Image Proc., vol. 8, no. 6, pages 879–885, June 1999.

[Shen 02] J. Shen & T. F. Chan. *Mathematical models for local nontexture inpaintings.* SIAM Journal on Applied Mathematics, vol. 62, pages 1019–1043, 2002.

[Shen 03] J. Shen, S. H. Kang & T. F. Chan. *Euler's elastica and curvature-based inpainting.* SIAM Journal on Applied Mathematics, vol. 63, pages 564–592, 2003.

[Shen 09] B. Shen, W. Hu, Y. Zhang & Y.-J. Zhang. *Image inpainting via sparse representation.* In Proceedings of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), pages 697–700, 2009.

[Simakov 08] D. Simakov, Y. Caspi, E. Shechtman & M. Irani. *Summarizing visual data using bidirectional similarity.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 1–8, 2008.

[Simoncelli 92] E. P. Simoncelli, W. T. Freeman, E. H. Adelson & D. J. Heeger. *Shiftable multiscale transforms.* IEEE Trans. on Information Theory, vol. 38, no. 2, pages 587–607, March 1992.

[Solanki 09] S. V. Solanki & A. R. Mahajan. *Cracks inspection and interpolation in digitized artistic picture using image processing approach.* International Journal of Recent Trends in Engineering (IJRTE), vol. 1, no. 2, pages 97–99, May 2009.

[Spagnolo 10] G. S. Spagnolo & F. Somma. *Virtual restoration of cracks in digitized image of paintings.* Journal of Physics: Conference Series, vol. 249, no. 1, page 012059, 2010.

[Stark 89] H. Stark & P. Oskoui. *High-resolution image recovery from image-plane arrays, using convex projections.* Journal of the Optical Society of America A (JOSA A), vol. 6, no. 11, pages 1715–1726, November 1989.

[Su 04] D. Su & P. Willis. *Image interpolation by pixel-level data-dependent triangulation.* Computer Graphics Forum, vol. 23, no. 2, pages 189–202, 2004.

[Suetake 08] N. Suetake, M. Sakano & E. Uchino. *Image super-resolution based on local self-similarity.* Optical Review, vol. 15, pages 26–30, 2008.

[Sun 03] J. Sun, N.-N. Zheng, H. Tao & H.-Y. Shum. *Image hallucination with primal sketch priors.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 729–736, 2003.

[Sun 05] J. Sun, L. Yuan, J. Jia & H.-Y. Shum. *Image completion with structure propagation.* ACM Trans. Graph., vol. 24, pages 861–868, July 2005.

[Sun 08] J. Sun, J. Sun, Z. Xu & H.-Y. Shum. *Image super-resolution using gradient profile prior.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 1–8, 2008.

[Sun 10] J. Sun, J. Zhu & M. F. Tappen. *Context-constrained hallucination for image super-resolution.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 231–238, 2010.

[Szeliski 08] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen & C. Rother. *A comparative study of energy minimization methods for Markov random fields with smoothness-based prior.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 30, no. 6, pages 1068–1080, June 2008.

[Tai 06] Y.-W. Tai, W.-S. Tong & C.-K. Tang. *Perceptually-inspired and edge-directed color image super-resolution.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 1948–1955, 2006.

[Tai 10] Y.-W. Tai, S. Liu, M. S. Brown & S. Lin. *Super resolution using edge prior and single image detail synthesis.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 2400–2407, 2010.

[Takeda 07] H. Takeda, S. Farsiu & P. Milanfar. *Kernel regression for image processing and reconstruction.* IEEE Trans. on Image Proc., vol. 16, no. 2, pages 349–366, February 2007.

[Tappen 03] M. F. Tappen, B. C. Russell & W. T. Freeman. *Exploiting the sparse derivative prior for super-resolution and image demosaicing.* In IEEE Workshop on Statistical and Computational Theories of Vision, 2003.

[Tappen 04] M. F. Tappen, B. C. Russell & W. T. Freeman. *Efficient graphical models for processing images.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 673–680, 2004.

[Tikhonov 77] A. N. Tikhonov & V. A. Arsenin. Solutions of Ill-posed Problems. Winston & Sons, Washington, 1977.

[Tom 95] B. C. Tom & A. K. Katsaggelos. *Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), volume 2, pages 539–542, 1995.

[Torralba 10] A. Torralba, K. P. Murphy & W. T. Freeman. *Using the forest to see the trees: exploiting context for visual object detection and localization.* Commun. ACM, vol. 53, no. 3, pages 107–114, March 2010.

[Torre 86] V. Torre & T. Poggio. *On edge detection.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 8, no. 2, pages 147–163, February 1986.

[Tsai 84] R. Y. Tsai & T. S. Huang. *Multiple frame image restoration and registration.* Advances in Computer Vision and Image Processing, vol. 1, pages 317–339, 1984.

[Tschumperlé 00] D. Tschumperlé & R. Deriche. *Vector-valued image regularization with PDE's: A common framework for different applications.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 27, no. 4, 200.

[Tschumperlé 06] D. Tschumperlé. *Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's.* Int. Journal of Computer Vision, vol. 68, no. 1, pages 65–82, June 2006.

[Tuceryan 90] M. Tuceryan & A. K. Jain. *Texture segmentation using Voronoi polygons.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 12, no. 2, pages 211–216, February 1990.

[Tuceryan 98] M. Tuceryan & A. K. Jain. *Texture analysis.* In C. H. Chen, L. F. Pau & P. S. P. Wang, editeurs, The Handbook of Pattern Recognition and Computer Vision, pages 207–248. World Scientific Publishing Co., 2nd edition, 1998.

[Unser 86] M. Unser. *Sum and difference histograms for texture classification.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 8, no. 1, pages 118–125, January 1986.

[Unser 90] M. Unser & M. Eden. *Nonlinear operators for improving texture segmentation based on features extracted by spatial filtering.* IEEE Trans. on Systems, Man and Cybernetics, vol. 20, no. 4, pages 804–815, July-August 1990.

[Unser 99] M. Unser. *Splines: A perfect fit for signal and image processing.* IEEE Sig. Proc. Magazine, vol. 16, no. 6, pages 22–38, November 1999.

[Unser 00] M. Unser. *Sampling - 50 years after Shannon.* Proceedings of the IEEE, vol. 88, no. 4, pages 569–587, April 2000.

[Ur 82] H. Ur & D. Gross. *Improved resolution from subpixel shifted pictures.* CVGIP: Graphical Models and Image Processing, vol. 54, no. 2, pages 181–186, March 1982.

[van der Maaten 10] L. J. P. van der Maaten & E. O. Postma. *Texton-based analysis of paintings.* In A. G. Tescher, editeur, Applications of Digital Image Processing XXXIII; Proceedings of SPIE, volume 7798, 2010.

[Varma 02] M. Varma & A. Zisserman. *Classifying images of materials: Achieving viewpoint and illumination ondependence.* In Proceedings of European Conf. on Computer Vision (ECCV), pages 255–271, 2002.

[Varma 03] M. Varma & A. Zisserman. *Texture classification: Are filter banks necessary?* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 691–698, 2003.

[Varma 05] M. Varma & A. Zisserman. *A statistical approach to texture classification from sngle images.* Int. Journal of Computer Vision, vol. 62, no. 1-2, pages 61–81, April 2005.

[Vidal 05] R. Vidal, Y. Ma & S. Sastry. *Generalized principal component analysis (GPCA).* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 27, no. 12, pages 1945–1959, December 2005.

[Voronin 11] V. V. Voronin, V. I. Marchuk & K. O. Egiazarian. *Images reconstruction using modified exemplar based method.* In Image Processing: Algorithms and Systems IX; Proceedings of SPIE, volume 7870, 2011.

[Voronin 12] V. V. Voronin, V. I. Marchuk, A. I. Sherstobitov & K. O. Egiazarian. *Image inpainting using cubic spline-based edge reconstruction.* In Image Processing: Algorithms and Systems X; and Parallel Processing for Imaging Applications II; Proceedings of SPIE, volume 8295, 2012.

[Wainwright 05] M. J. Wainwright, T. S. Jaakkola & A. S. Willsky. *MAP estimation via agreement on trees: message-passing and linear programming.* IEEE Trans. on Information Theory, vol. 51, no. 11, pages 3697–3717, November 2005.

[Walden 85] S. Walden. The Ravished Image. St. Martin's Press, New York, 1985.

[Wang 04] Z. Wang, A. C. Bovik, H. R. Sheikh & E. P. Simoncelli. *Image quality assessment: From error visibility to structural similarity.* IEEE Trans. on Image Proc., vol. 13, no. 4, pages 600–612, April 2004.

[Wang 05] Q. Wang, X. Tang & H. Shum. *Patch based blind image super resolution.* In Proceedings of Int. Conf. on Computer Vision (ICCV), pages 709–716, 2005.

[Wang 07] Q. Wang & R. K. Ward. *A new orientation-adaptive interpolation method.* IEEE Trans. on Image Proc., vol. 16, no. 4, pages 889–900, April 2007.

[Wang 10] J. Wang, S. Zhu & Y. Gong. *Resolution enhancement based on learning the sparse association of image patches.* Pattern Recognition Letters, vol. 31, no. 1, pages 1–10, January 2010.

[Wei 00] L.-Y. Wei & M. Levoy. *Fast texture synthesis using tree-structured vector quantization.* In SIGGRAPH, pages 479–488, 2000.

[Weldon 96] T. P. Weldon, W. E. Higgins & D. F. Dunn. *Efficient Gabor filter design for texture segmentation.* Pattern Recognition, vol. 29, no. 12, pages 2005–2015, December 1996.

[Wexler 07] Y. Wexler, E. Shechtman & M. Irani. *Space-time completion of video.* IEEE Trans. on Pattern Anal. and Machine Intel., vol. 29, no. 3, pages 463–476, March 2007.

[Williams 95] L. R. Williams & D. W. Jacobs. *Stochastic completion fields: A neural model of illusory contour shape and salience.* In Proceedings of Int. Conf. on Computer Vision (ICCV), pages 408–415, 1995.

[Winkler 95] G. Winkler. Image Analysis, Random Fields, and Dynamic Monte Carlo Methods. Springer-Verlag, 1995.

[Won 04] C. S. Won & R. M. Gray. Stochastic Image Processing. Kluwer Academic/Plenum Publishers, New York, 2004.

[Wong 08] A. Wong & J. Orchard. *A nonlocal-means approach to exemplar-based inpainting.* In Proceedings of IEEE Int. Conf. on Image Processing (ICIP), pages 2600–2603, 2008.

[Xiong 09] Z. Xiong, X. Sun & F. Wu. *Image hallucination with feature enhancement.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 2074–2081, 2009.

[Xu 10] Z. Xu & J. Sun. *Image inpainting by patch propagation using patch sparsity.* IEEE Trans. on Image Proc., vol. 19, no. 15, pages 1153–1165, May 2010.

[Yang 08] J. Yang, J. Wright, T. S. Huang & Y. Ma. *Image super-resolution as sparse representation of raw image patches.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2008.

[Yang 09] Y. Yang, Y. Zhu & Q. Peng. *Image completion using structural priority belief propagation.* In Proceedings of ACM Int. Conf. on Multimedia, pages 717–720, 2009.

[Yang 10a] C.-Y. Yang, J.-B. Huang & M.-H. Yang. *Exploiting self-similarities for single frame super-resolution.* In Proceedings of Asian Conf. on Computer vision (ACCV), pages 497–510, 2010.

[Yang 10b] J. Yang, J. Wright, T. S. Huang & Y. Ma. *Image super-resolution via sparse representation.* IEEE Trans. on Image Proc., vol. 19, no. 11, pages 2861–2873, November 2010.

[Yedidia 00] J. S. Yedidia, W. T. Freeman & Y. Weiss. *Generalized belief propagation.* In Neural Information Processing Systems 13, pages 689–695, 2000.

[Yedidia 01a] J. S. Yedidia & W. T. Freeman. *On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs.* IEEE Trans. on Information Theory, vol. 47, no. 2, pages 736–744, February 2001.

[Yedidia 01b] J. S. Yedidia, W. T. Freeman & Y. Weiss. *Characterization of belief propagation and its generalizations.* Rapport technique, Mitsubishi Electric Research Laboratories, 2001.

[Yedidia 05] J.S. Yedidia, W.T. Freeman & Y. Weiss. *Constructing free-energy approximations and generalized belief propagation algorithms.* IEEE Trans. on Information Theory, vol. 51, no. 7, pages 2282–2312, July 2005.

[Young 85] R. A. Young. *The Gaussian derivative theory of spatial vision: Analysis of cortical cell receptive field line-weighting profiles.* Rapport technique, General Motors Research, 1985.

[Yu 01] X. Yu, B. S. Morse & T. W. Sederberg. *Image reconstruction using data-dependent triangulation.* IEEE Computer Graphics and Applications, vol. 21, pages 62–68, May 2001.

[Zalesny 01] A. Zalesny & L. J. Van Gool. *A compact model for viewpoint dependent texture synthesis.* In Revised Papers from European Workshop on 3D Structure from Multiple Images of Large-Scale Environments (SMILE), pages 124–143, 2001.

[Zana 01] F. Zana & J. C. Klein. *Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation.* IEEE Trans. on Image Proc., vol. 10, no. 7, pages 1010–1019, July 2001.

[Zeyde 10] R. Zeyde, M. Elad & M. Protter. *On single image scale-up using sparse-representations.* In Proceedings of Int. Conf. on Curves and Surfaces (ICCS), pages 711–730, 2010.

[Zhang 93] H. Zhang. *Image restoration: flexible neighborhood systems and iterated conditional expectations.* Statistica Sinica, vol. 3, pages 117–139, 1993.

[Zhu 98] S. C. Zhu, Y. N. Wu & D. Mumford. *Filters, Random Fields and Maximum Entropy (FRAME): Towards a unified theory for texture modeling.* Int. Journal of Computer Vision, vol. 27, no. 2, pages 107–126, March 1998.

[Zhu 05] S. C. Zhu, C. E. Guo, Y. Wang & Z. Xu. *What are textons?* Int. Journal of Computer Vision, vol. 62, no. 1-2, pages 121–143, April 2005.

[Zibetti 07] M. V. W. Zibetti & J. Mayer. *A robust and computationally efficient simultaneous super-resolution scheme for image sequences.* IEEE Trans. on Circuits and Systems for Video Technology, vol. 17, no. 10, pages 1288–1300, October 2007.

[Zomet 01] A. Zomet, A. Rav-acha & S. Peleg. *Robust super-resolution.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 645–650, 2001.

[Zontak 11] M. Zontak & M. Irani. *Internal statistics of a single natural image.* In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pages 977–984, 2011.