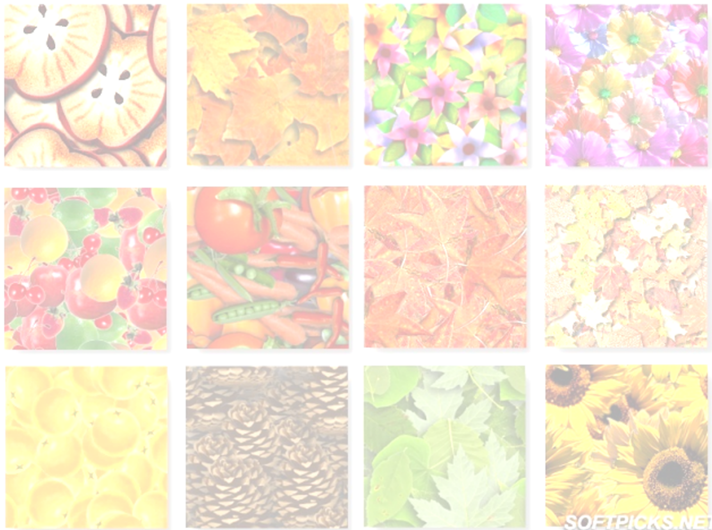
FACULTY OF ENGINEERING
AND ARCHITECTURE

E016712: Computer Graphics

Texture Synthesis

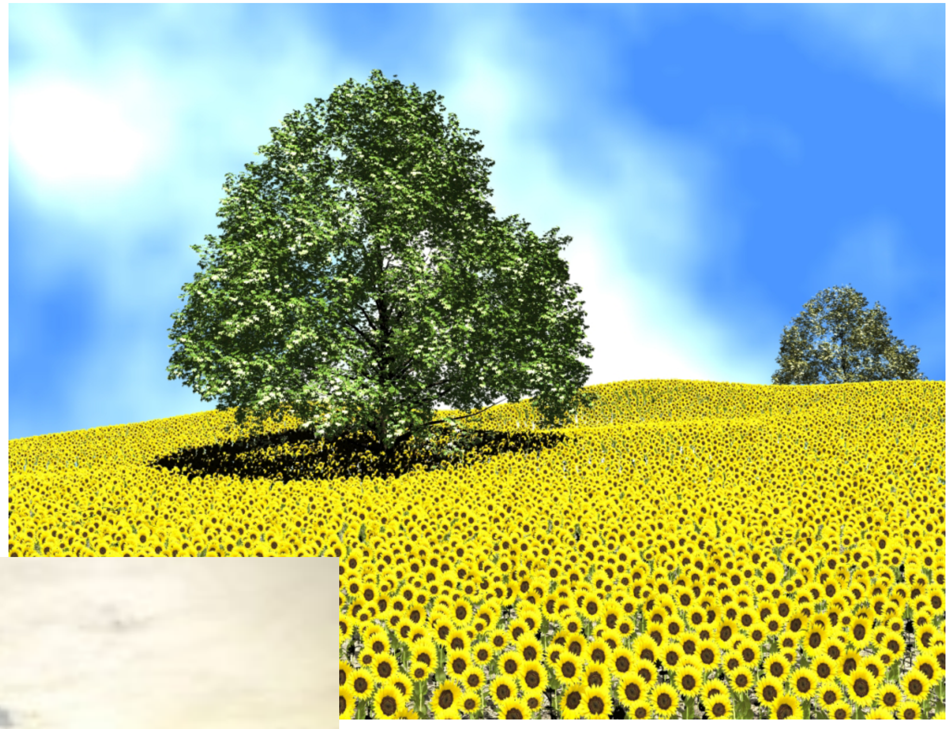


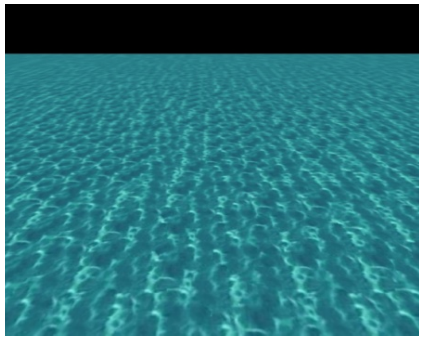
Lecturers: Aleksandra Pizurica and Danilo Babin

GHENT
UNIVERSITY

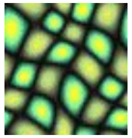
Applications of texture synthesis

- Computer generated images
- Films
- Virtual reality
- Computer games

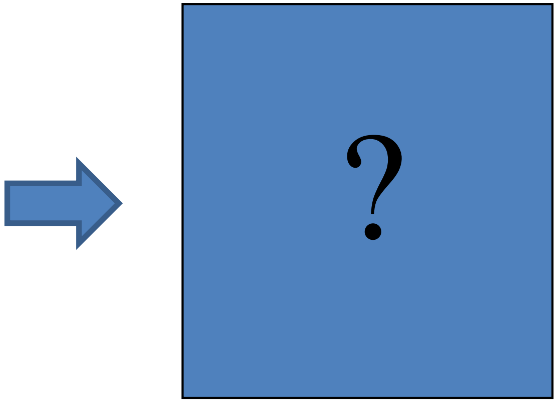




The problem

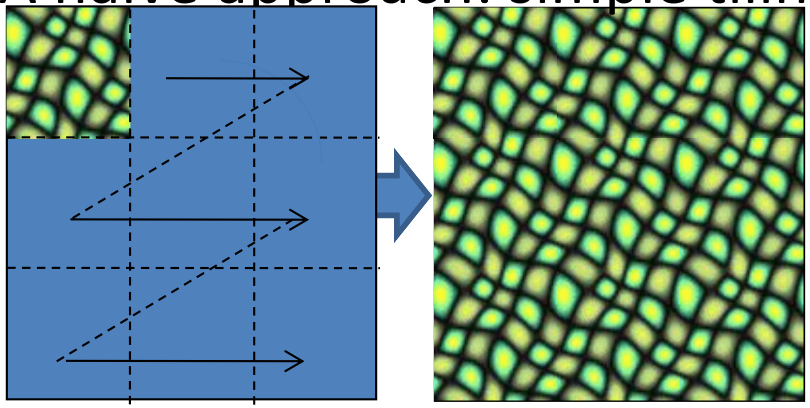


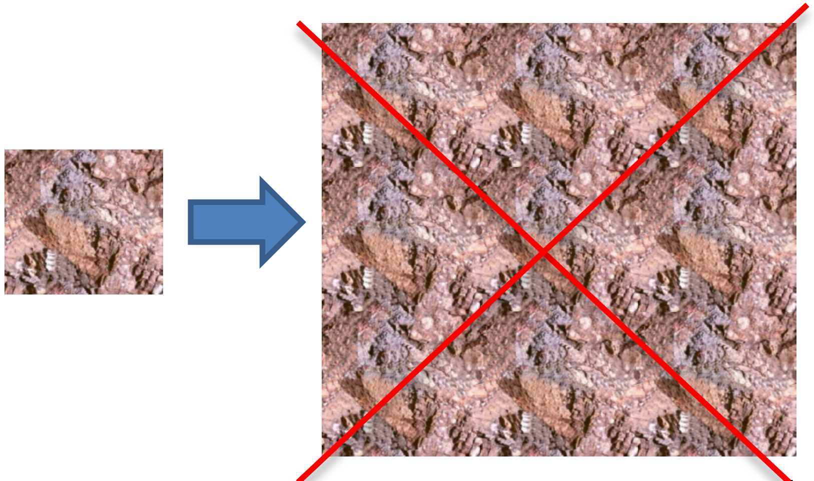
A sample



Large texture

A naive approach: simple tiling





Tiling is obvious!

Given texture *I*, generate texture *J*

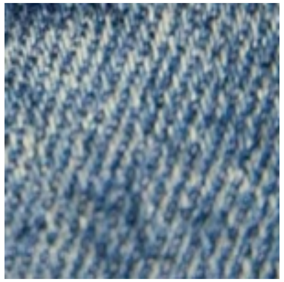
- That looks like *I* (or differs from *I* in the same way as *I* differs from itself)
- Doesn’t show any obvious copying or tiling

Types of Textures and Techniques

Deterministic
(well-defined primitives and placement rule)

Stochastic
(not easily identifiable primitives)

Mixture of the two
(most textures in practice)

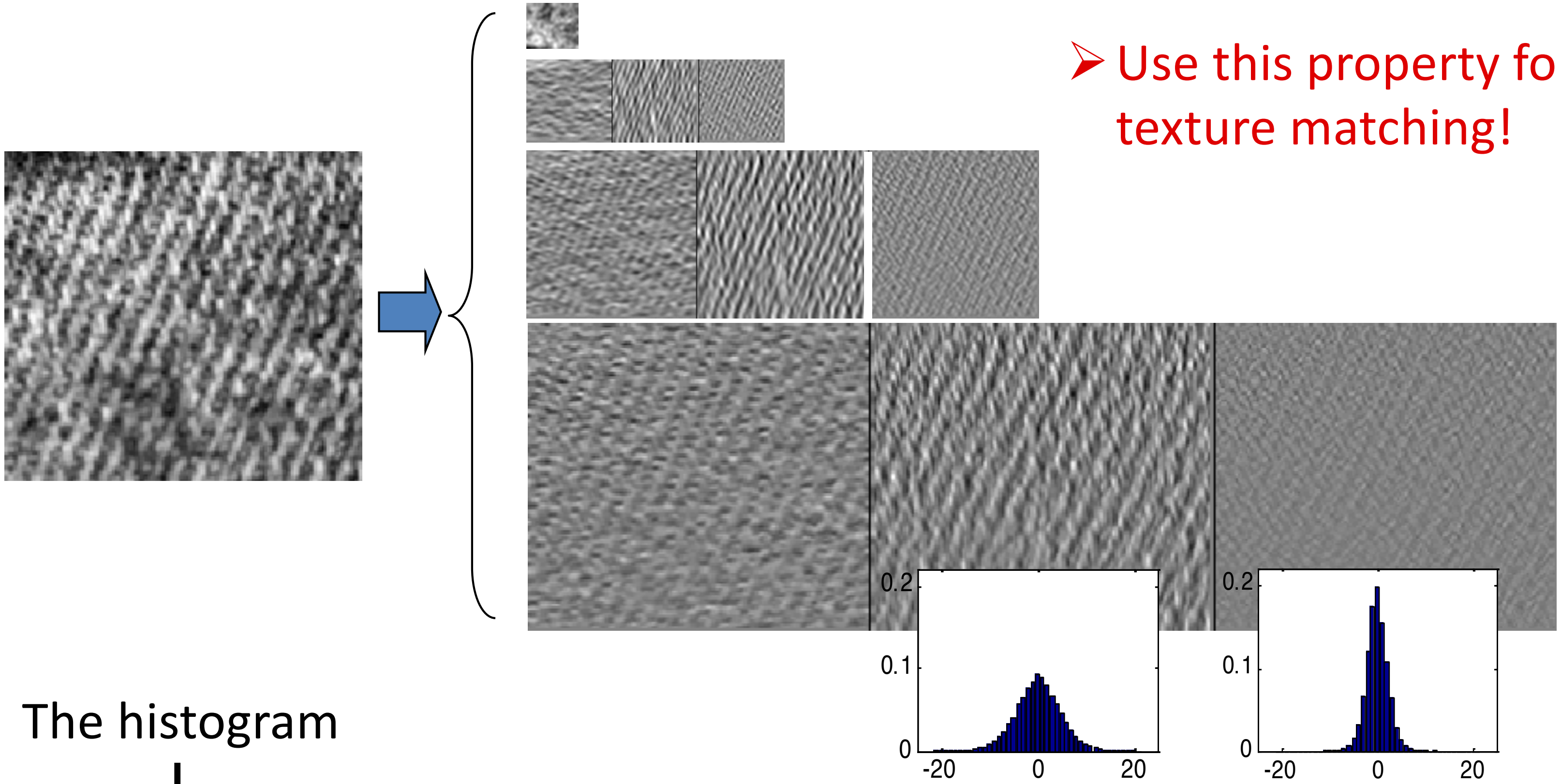


- **Model-based methods**
  - First-order statistics (pyramidal methods)
  - Using Markov Random Fields (We don't treat in this course)
- **Tiling and patch methods**
  - "Image Quilting"
  - Wang tiles
- **Hybrid methods** (tiling + stochastic modeling)

Pyramidal Methods

Pyramidal Methods (Heeger&Bergen)

Two textures are difficult to discriminate if they produce similar responses in a bank of (orientation and frequency selective) linear filters



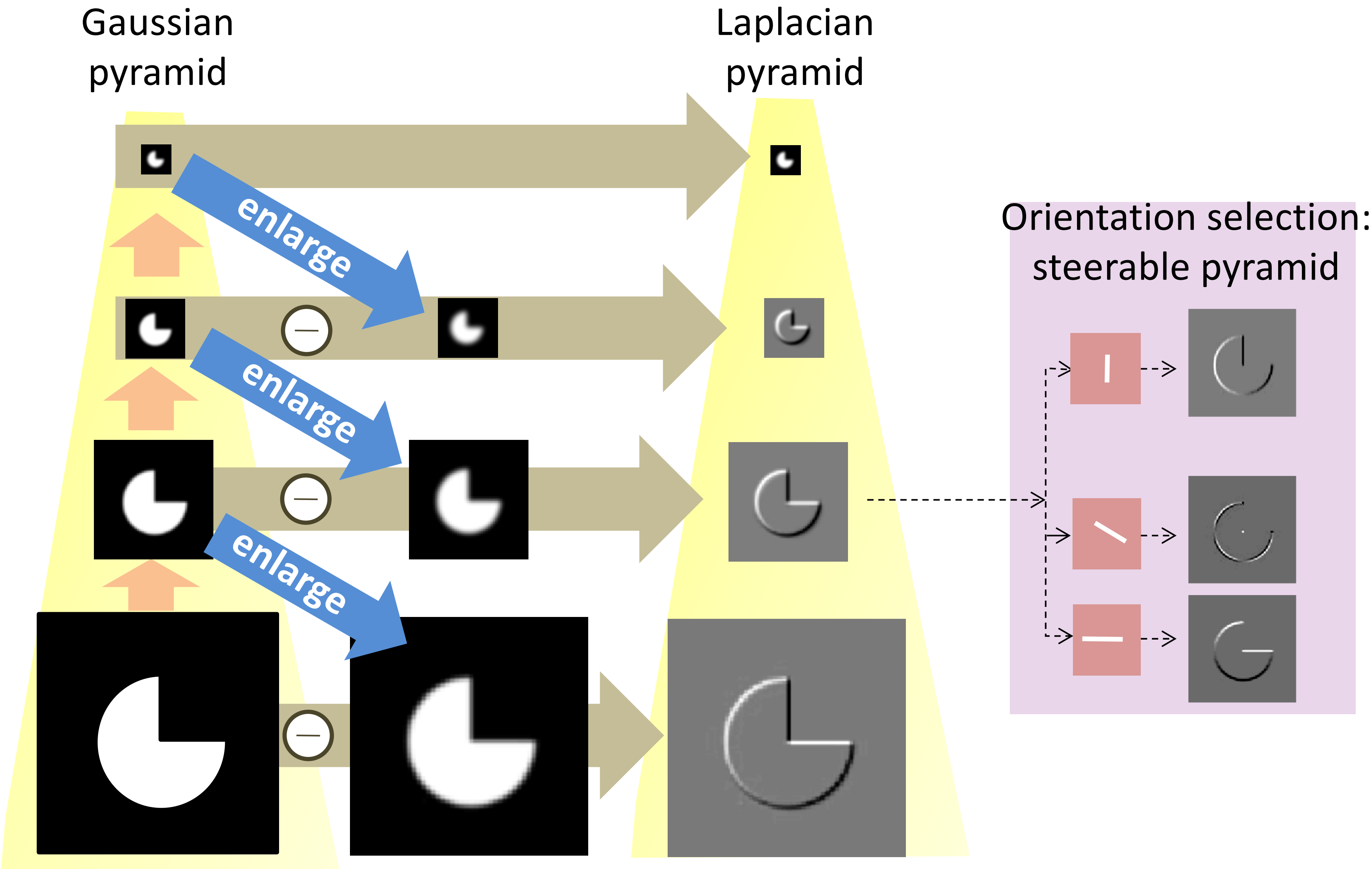
➢ Use this property for texture matching!

The histogram

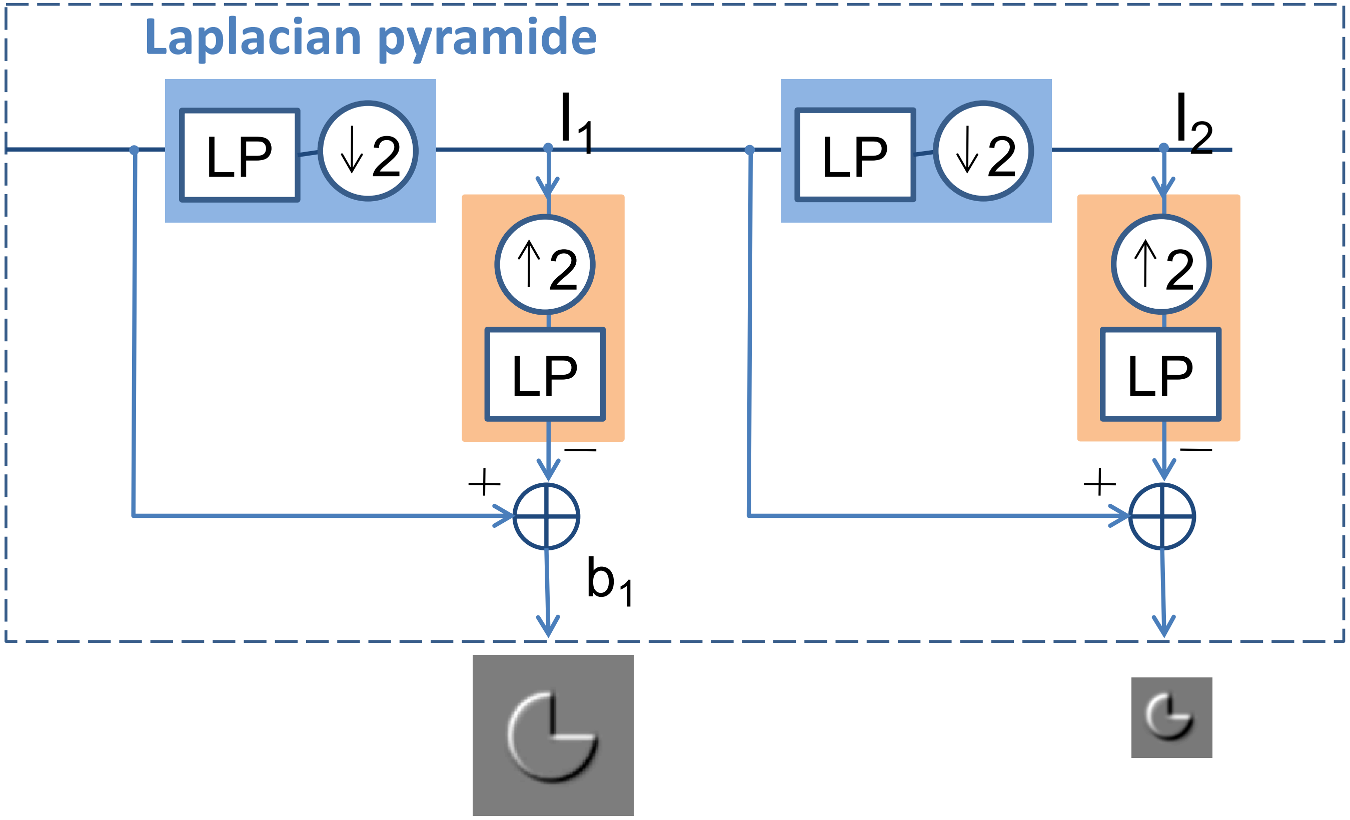
↓

Long tails→ more large coefficients → stronger edges in the corresponding subband (of a given orientation and resolution)

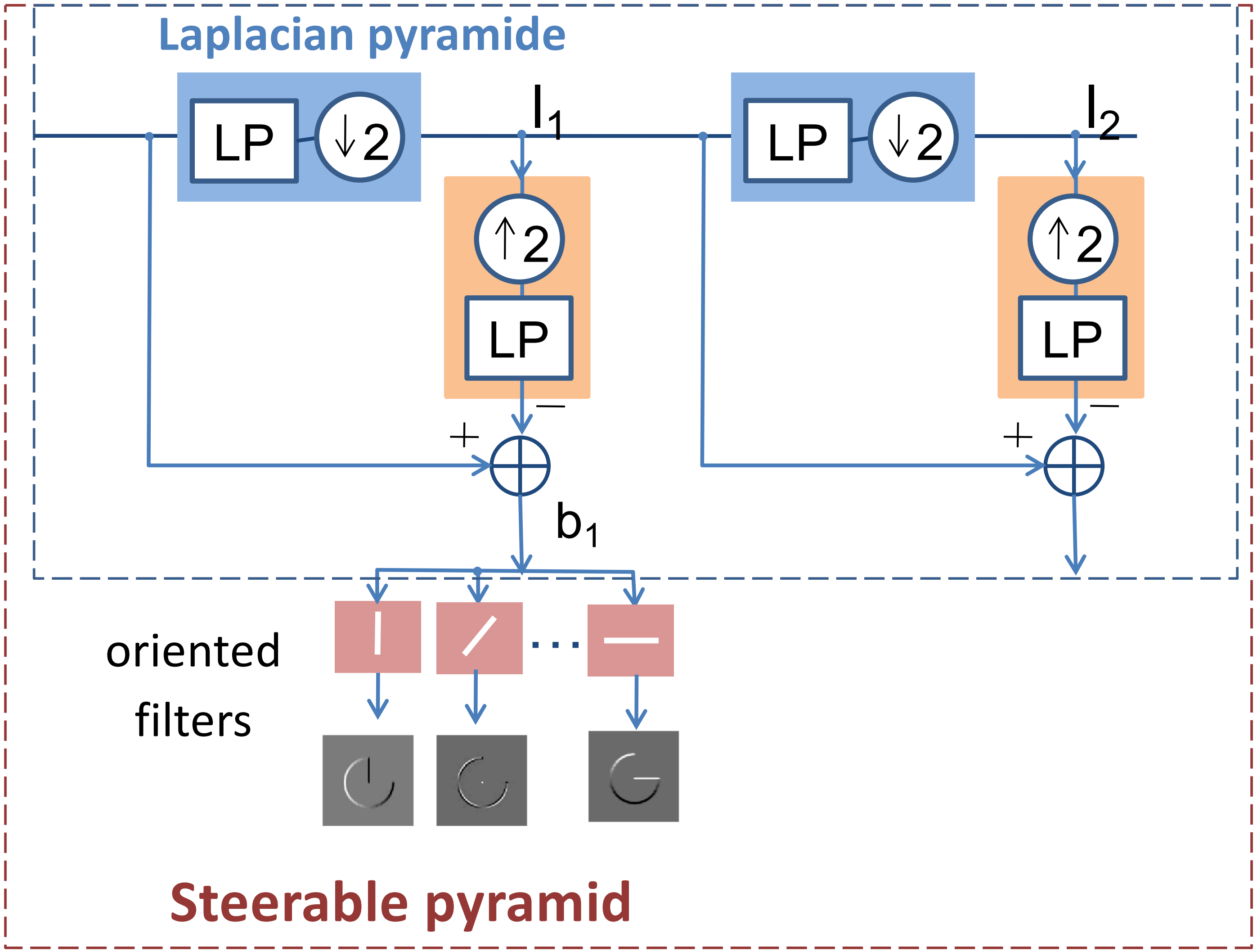
Pyramidal Decomposition



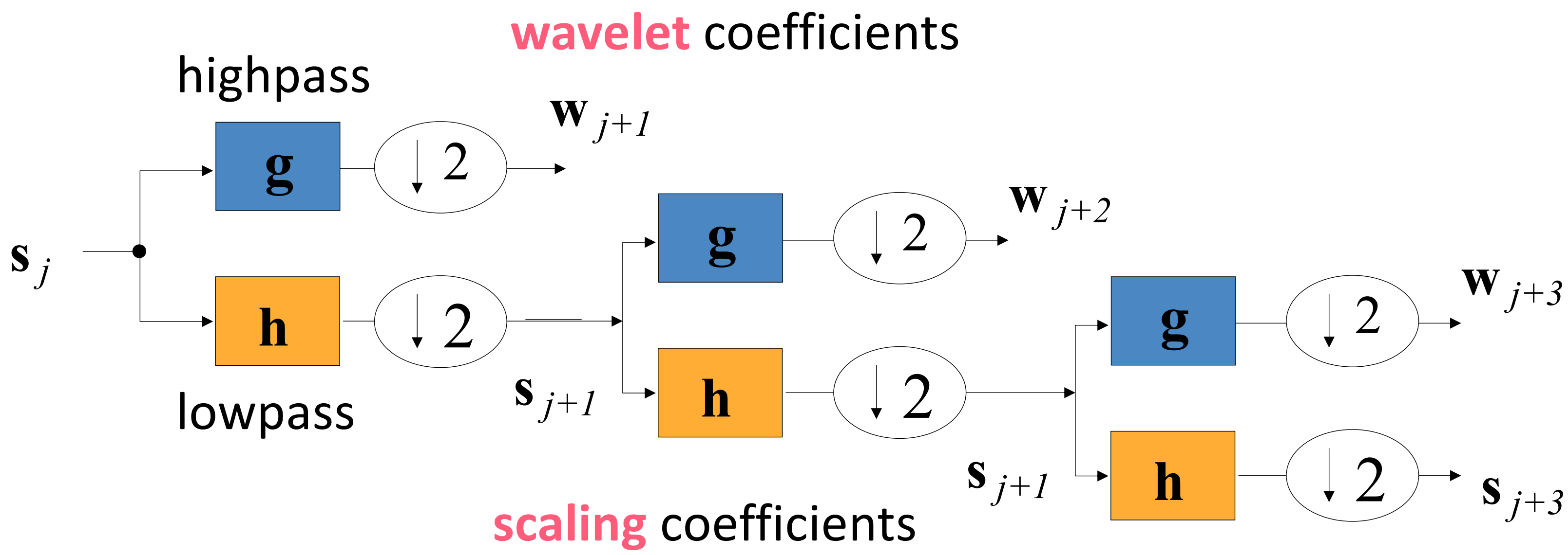
Pyramidal Decomposition



Pyramidal Decomposition

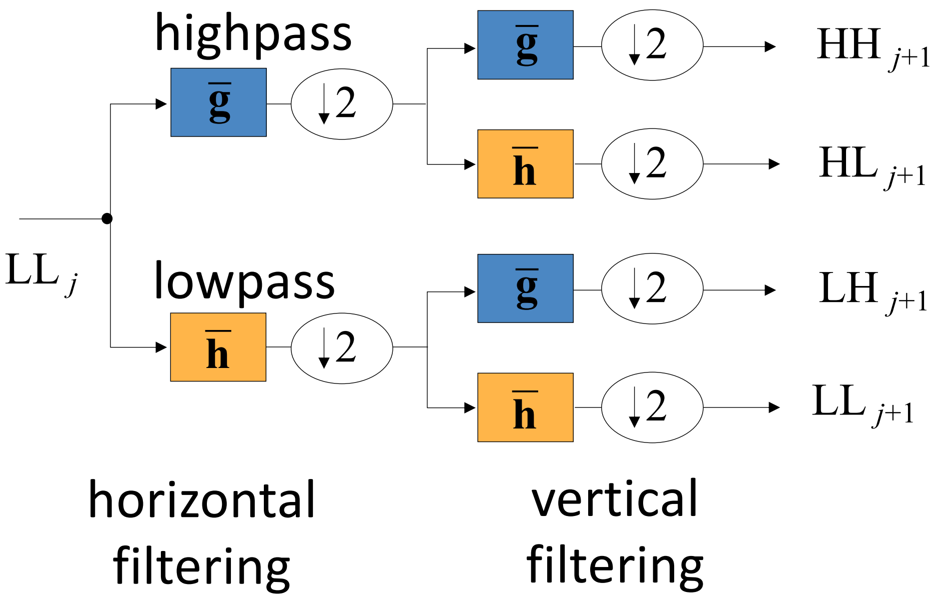


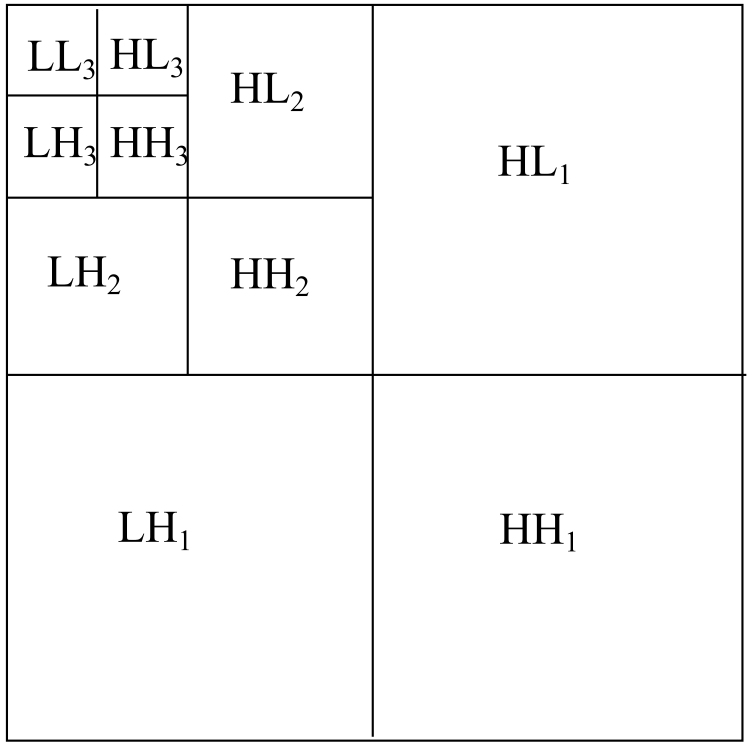
Discrete Wavelet Transform (DWT)



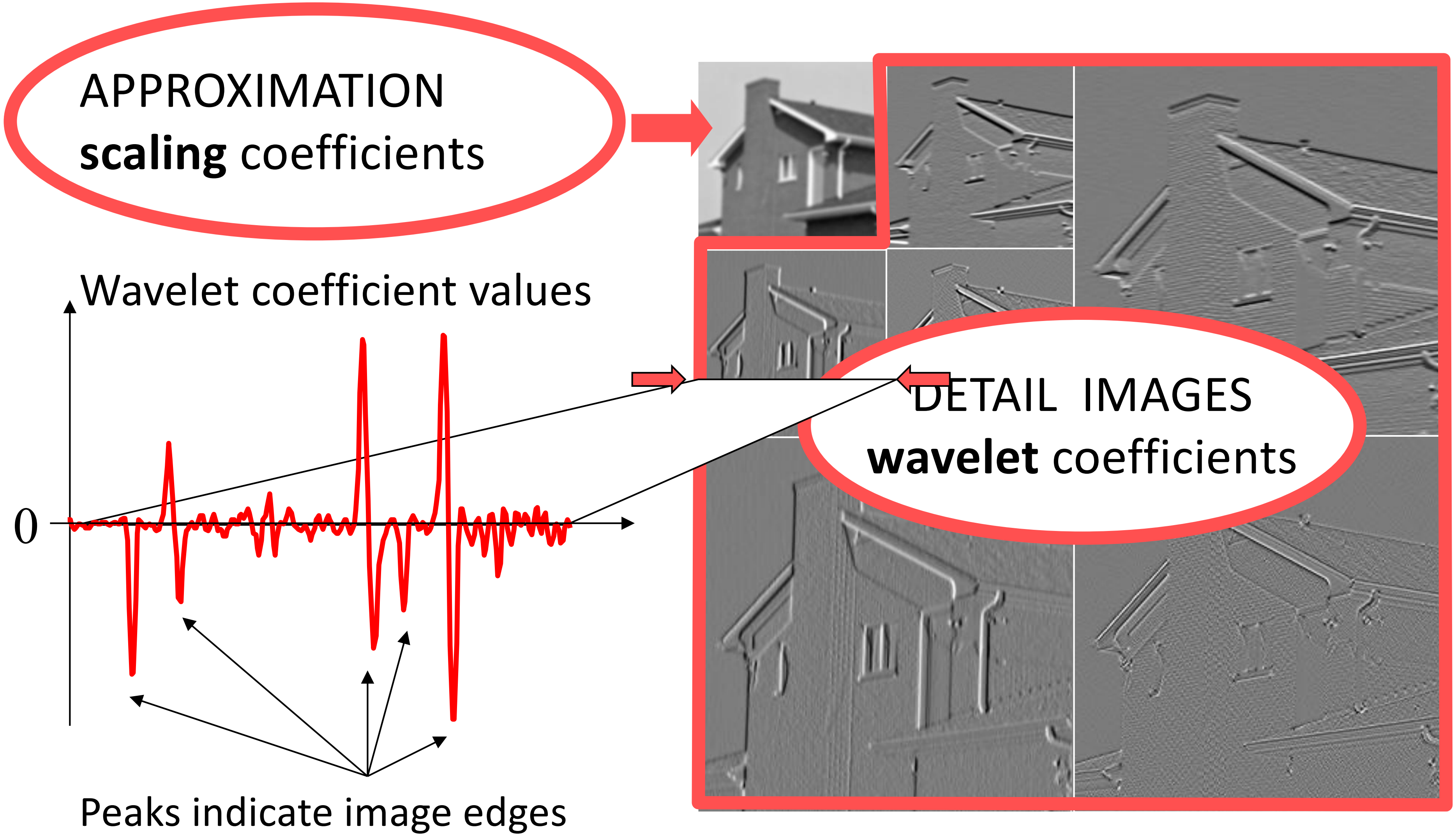
- DWT algorithm: a **filter bank** iterated on the lowpass output
- The same texture synthesis concept can be applied using image pyramid or DWT or a related multiresolution image representation

Two dimensional DWT

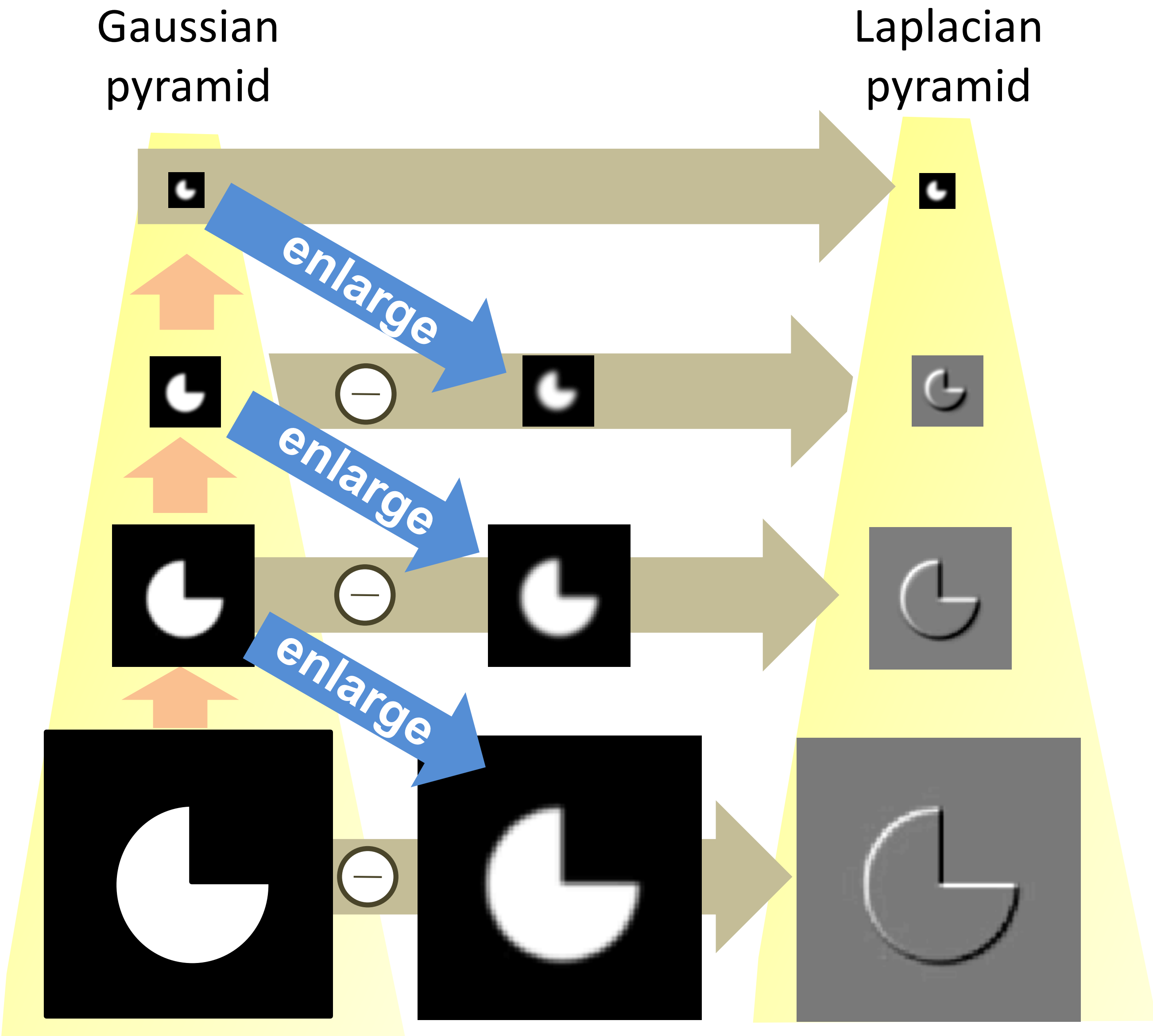




Two dimensional DWT

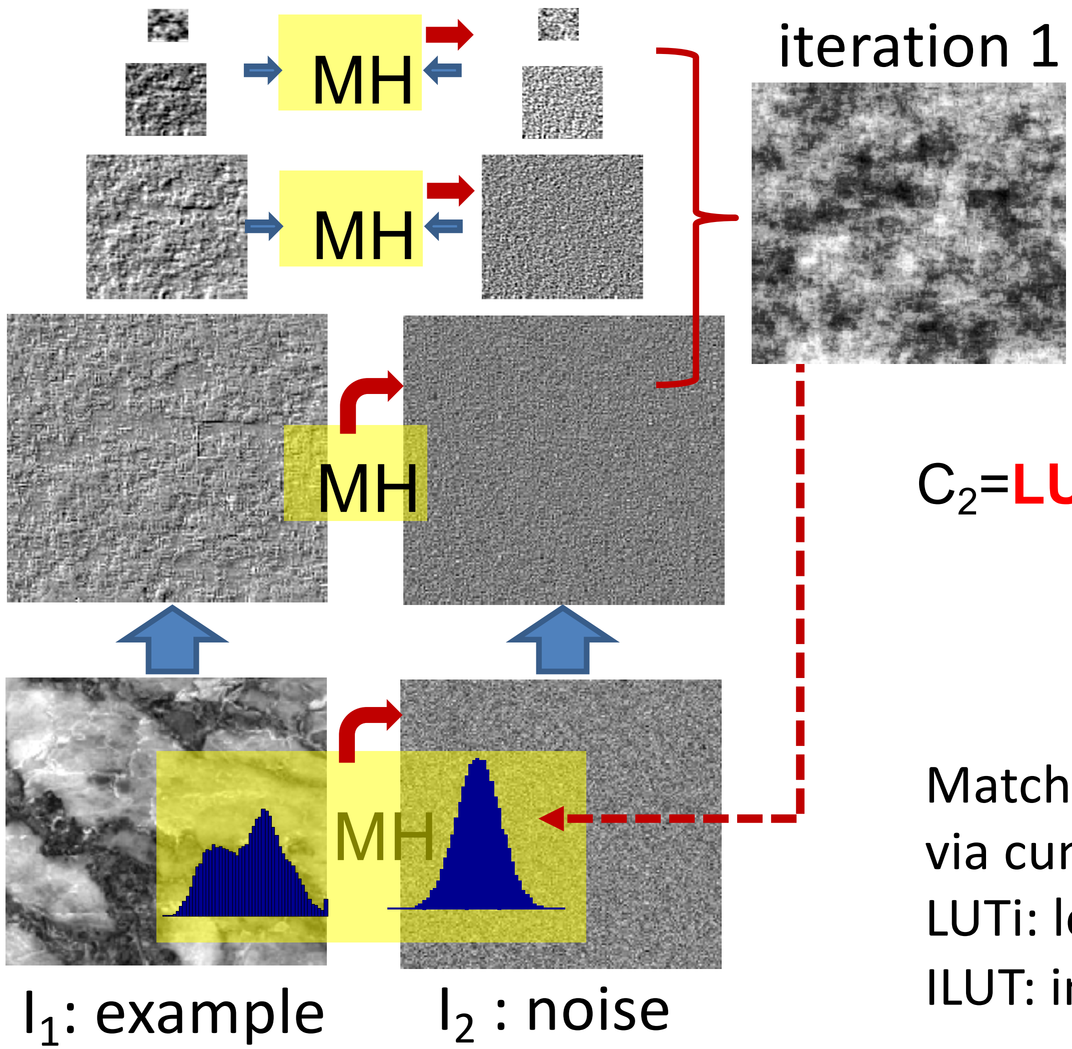


Texture Synthesis in Pyramidal Decomposition

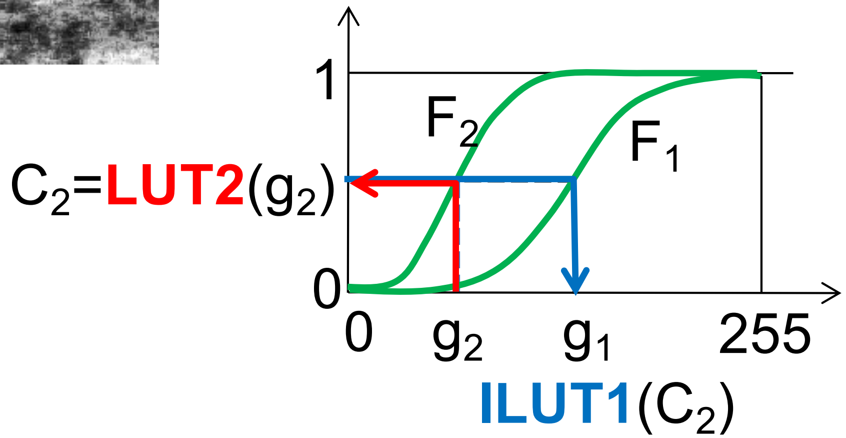


In the following, we assume a Laplacian pyramid, having on mind that the same approach can be applied in other multiresolution representations

Histogram Matching



$I_2(m,n)=$**ILUT1**(**LUT2**($I_2(m,n)$))

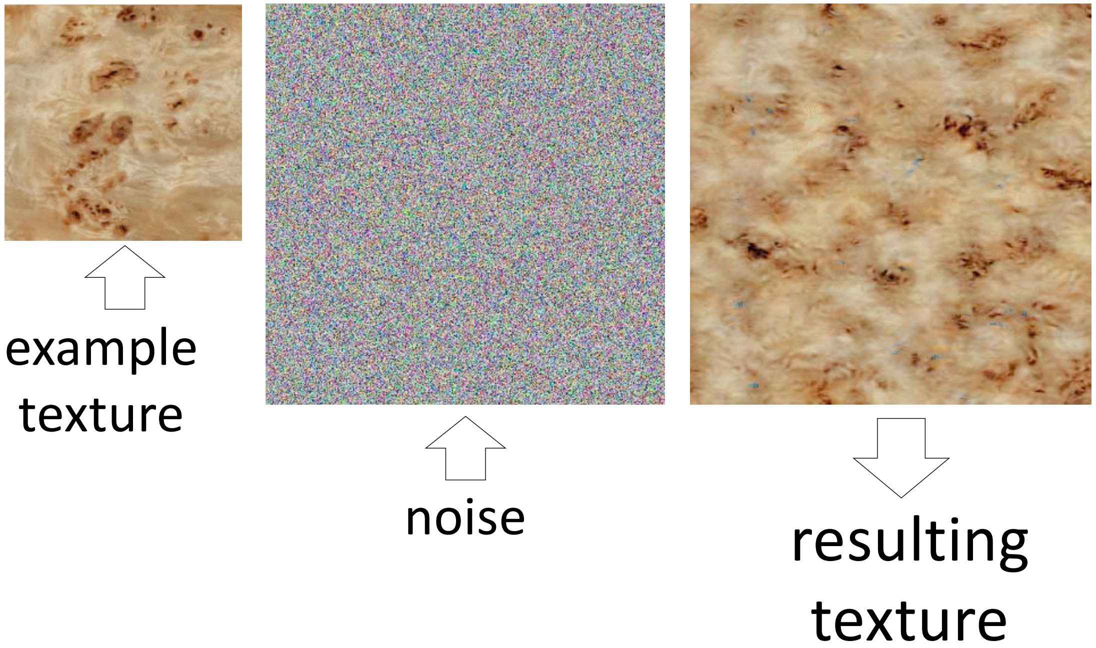


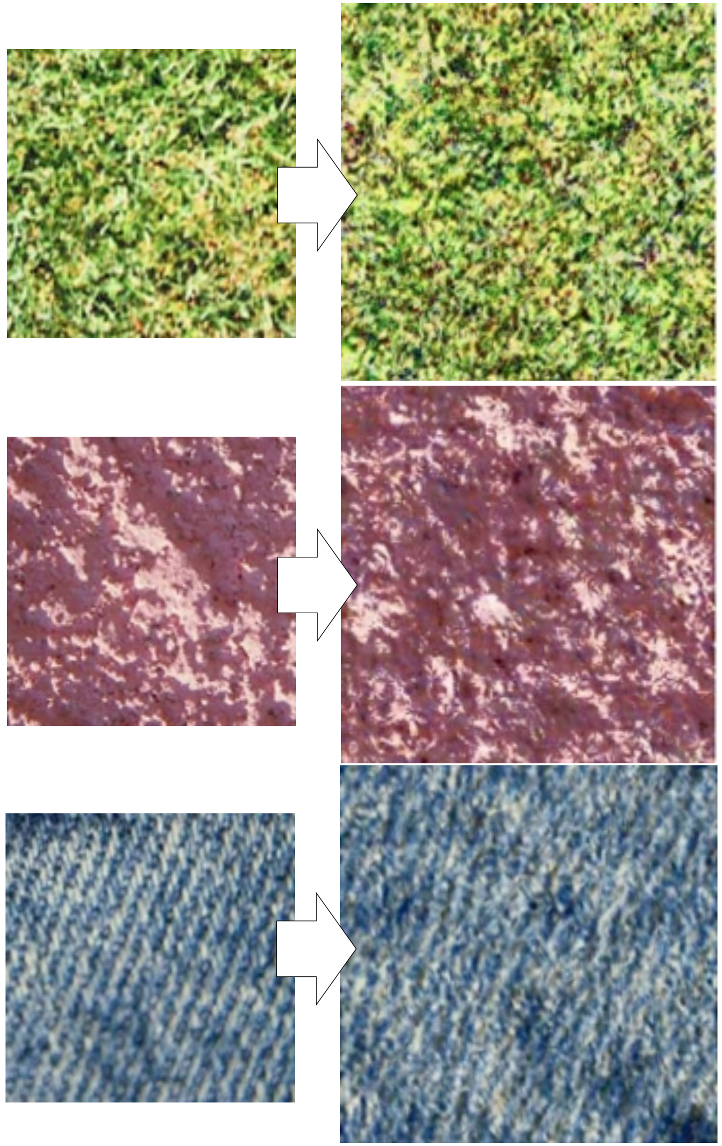
Matching histograms (MH)
via cumulative distributions F_1 and F_2.
LUTi: look-up table for F_i;
ILUT: inverse LUT.

When do we need more orientations?

Think of extension to color!

Pyramidal Methods – where they work?

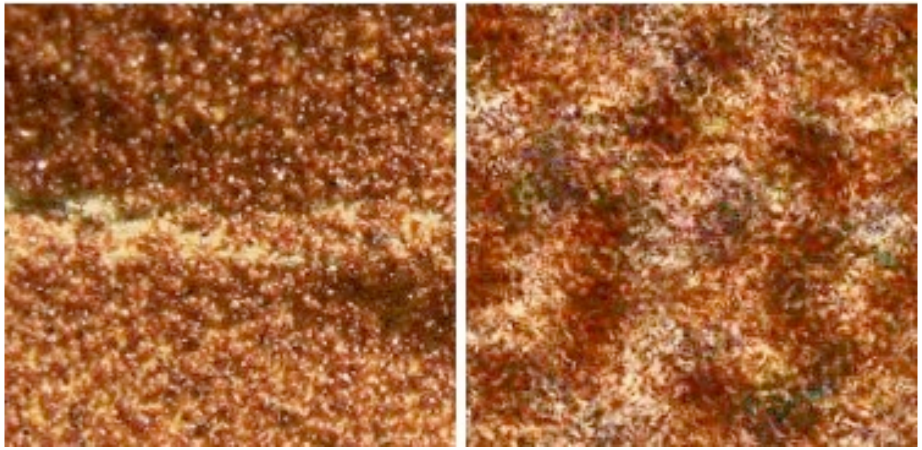




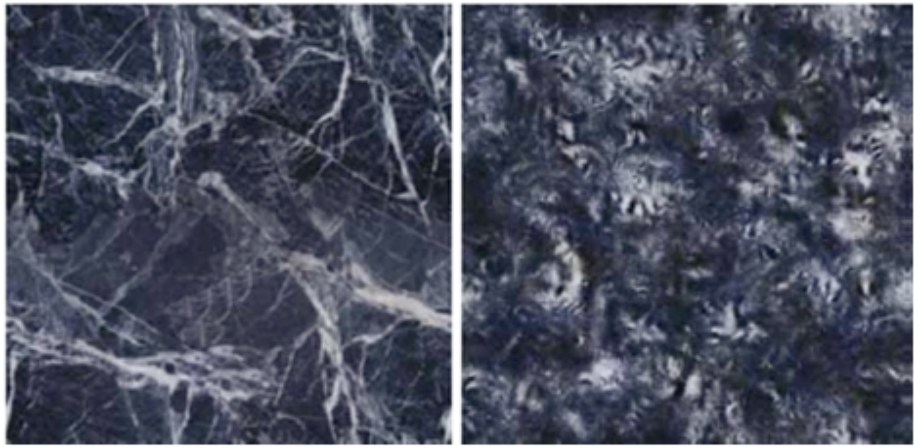
Suitable for homogeneous stochastic textures!

Pyramidal Methods – where they don't work?

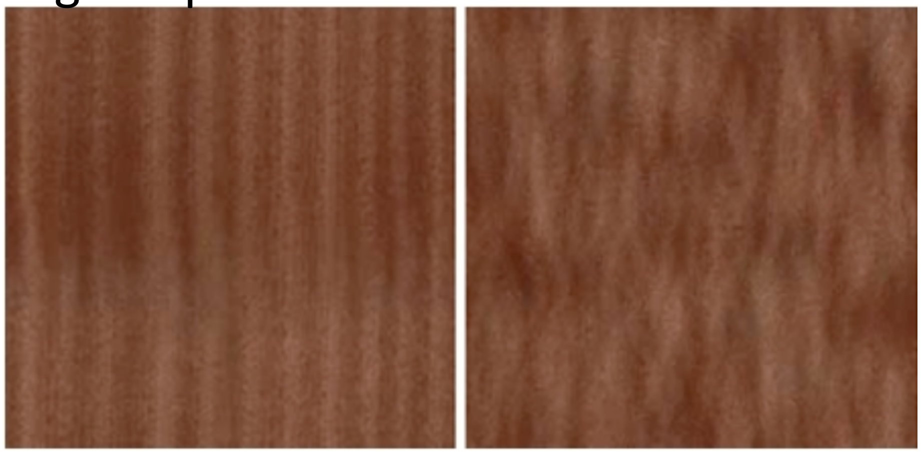
Not homogeneous texture



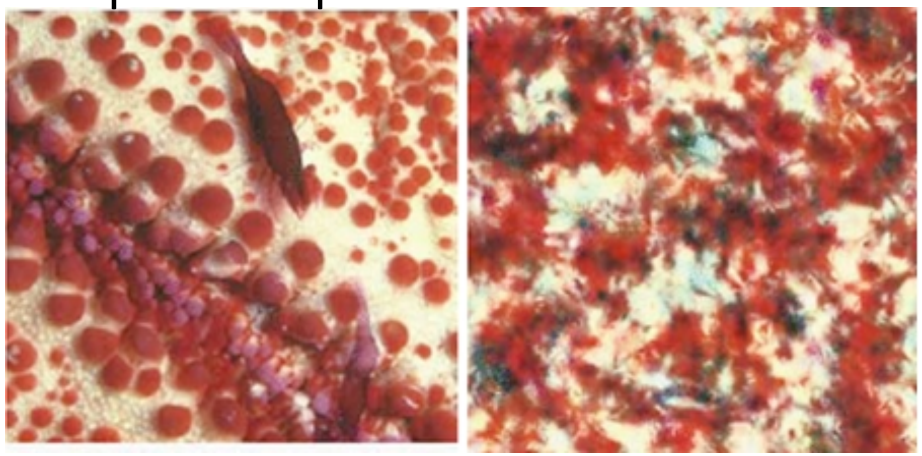
Characteristic structural details



Regular patterns



Complex compositions



The first-order statistics is not sufficient in these cases.

Pyramidal Methods: Summary

- Based on first-order statistics
- Motivated by studies of the human visual system
- Effective for homogeneous stochastic textures
- Reading:
  - D. J. Heeger and J.R. Bergen, “Pyramid Based Texture Analysis/Synthesis,” International Conference on Image Processing (ICIP'95) – Vol. 3, 1995.

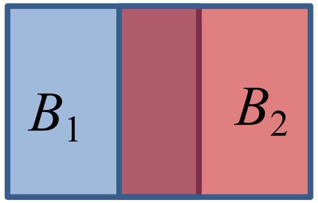
  → All of it

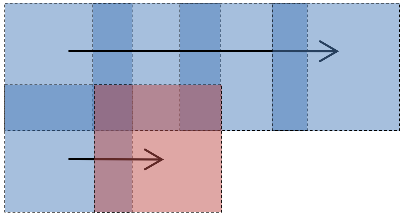
Image Quilting

Image Quilting (Efros&Freeman)

Choose blocks of a given size from the input texture sample

Step 1: Position **overlapping blocks sequentially**





Step 2: Calculate the **optimal borders in overlapping regions**

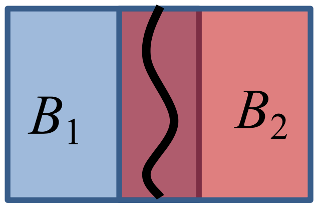


Image Quilting (Efros&Freeman)

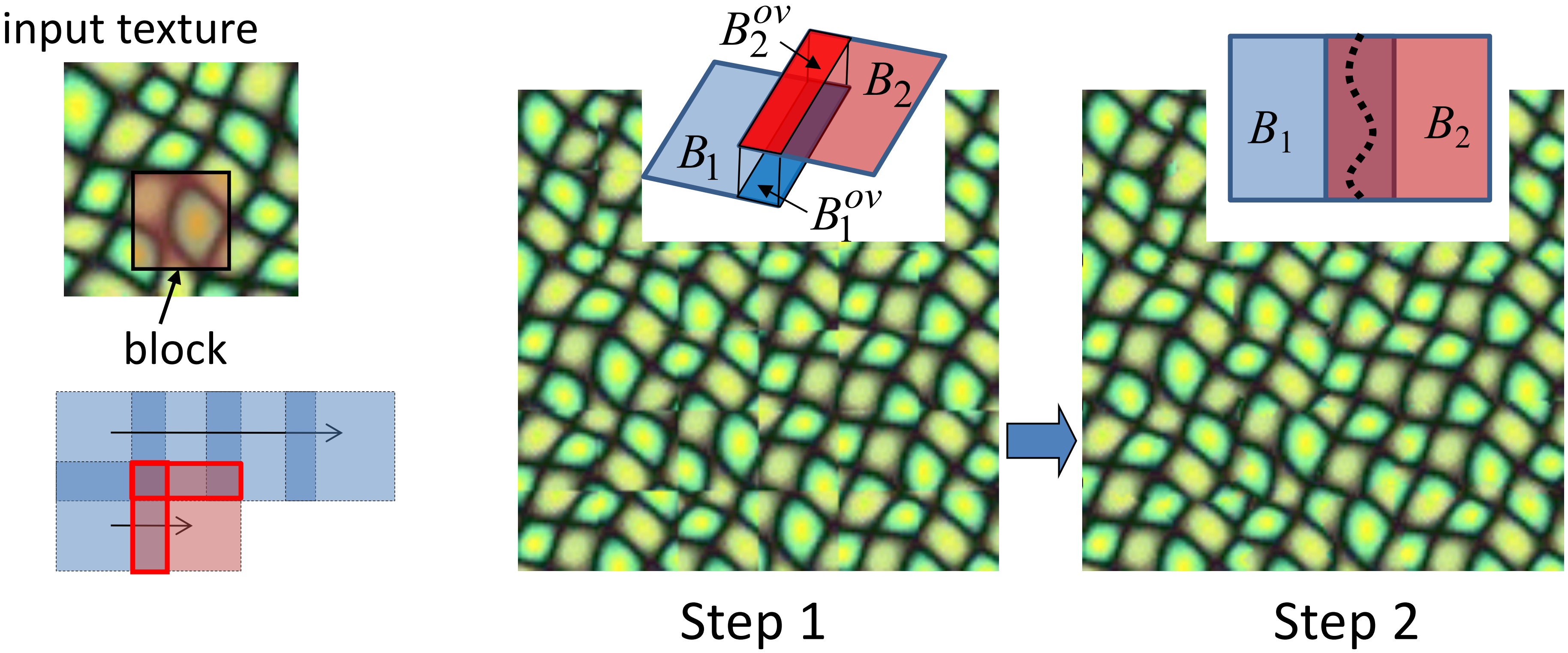
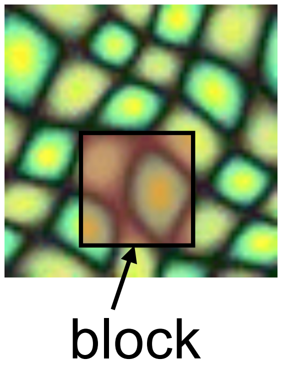
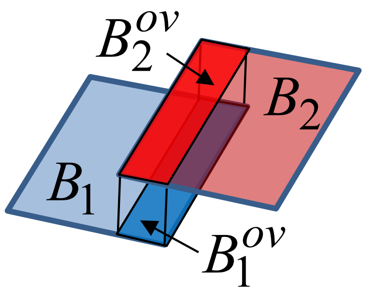


Image Quilting: Step 1

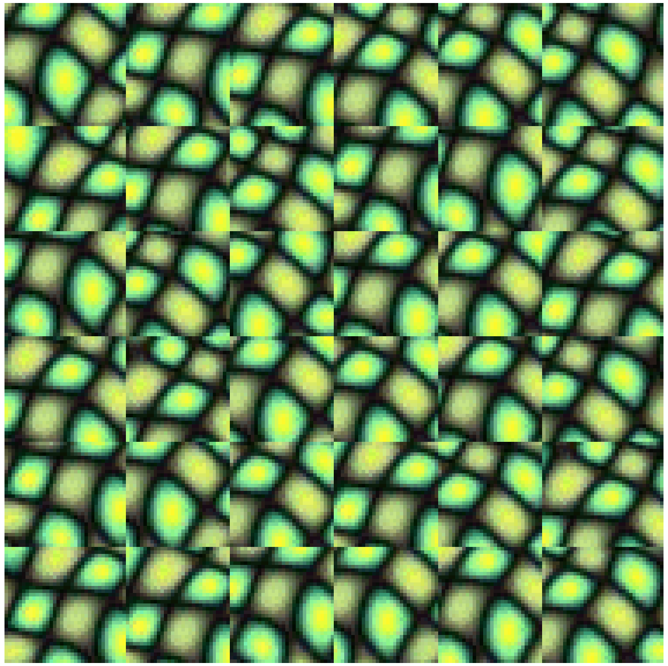
Position overlapping blocks sequentially

input

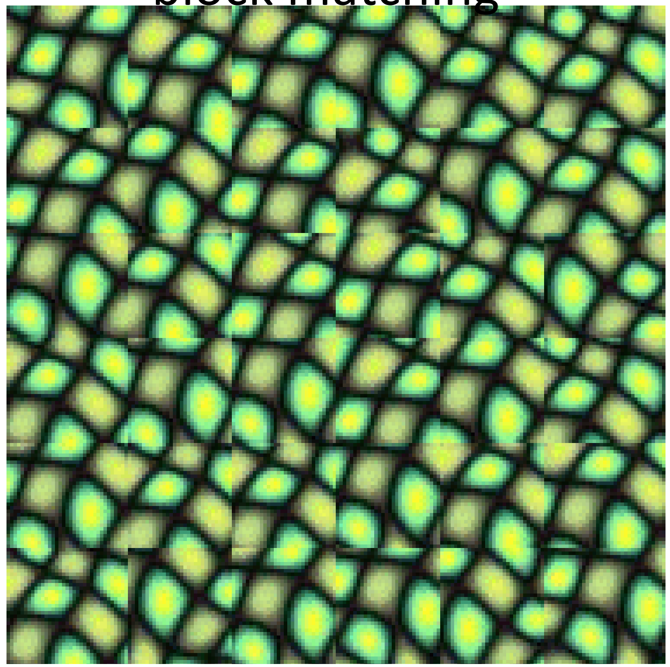




Random choice of blocks



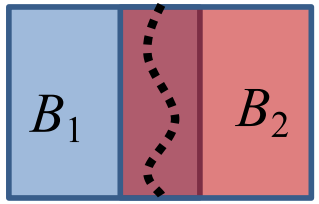
Beter: semi-random with "block matching"



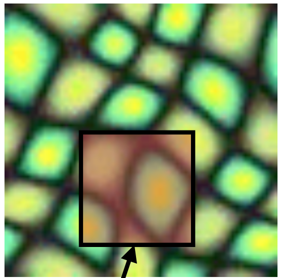
$$F_{block\,matching} = \sum_{i,j} | B_1^{ov}(i,j) - B_2^{ov}(i,j) | < Const$$

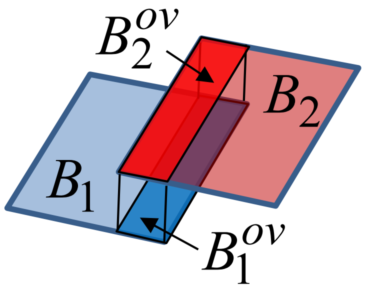
Image Quilting: Step 2

Find optimal borders in de overlapping regions

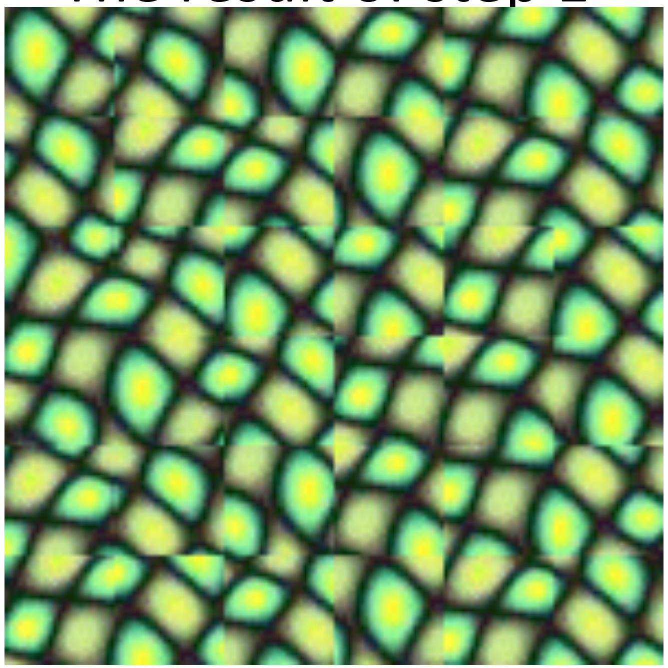


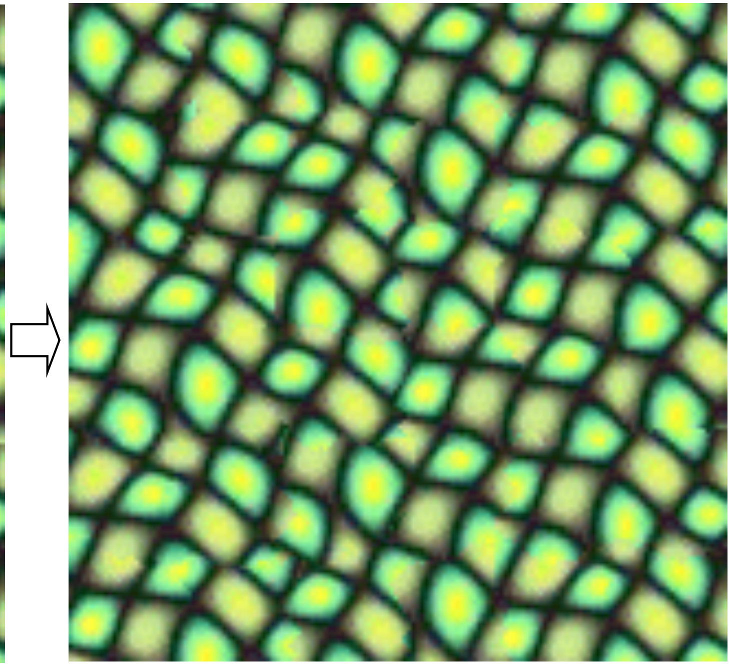
input





The result of step 1

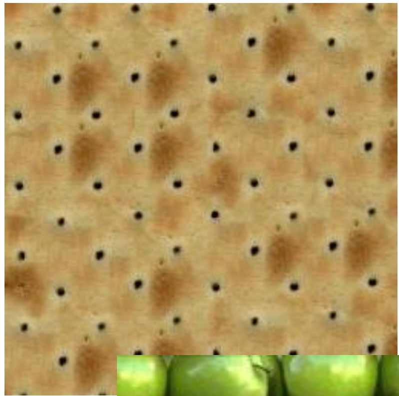


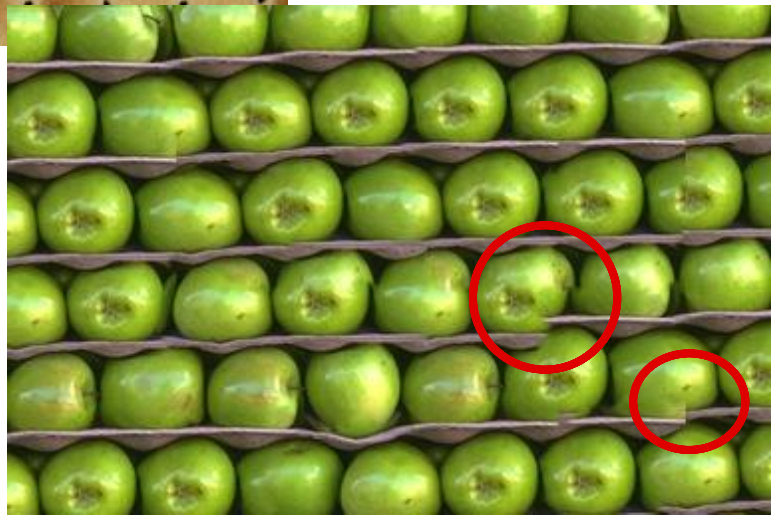


Shortest path through quadratic error surface

$$e(i,j) = (B_1^{ov}(i,j) - B_2^{ov}(i,j))^2$$

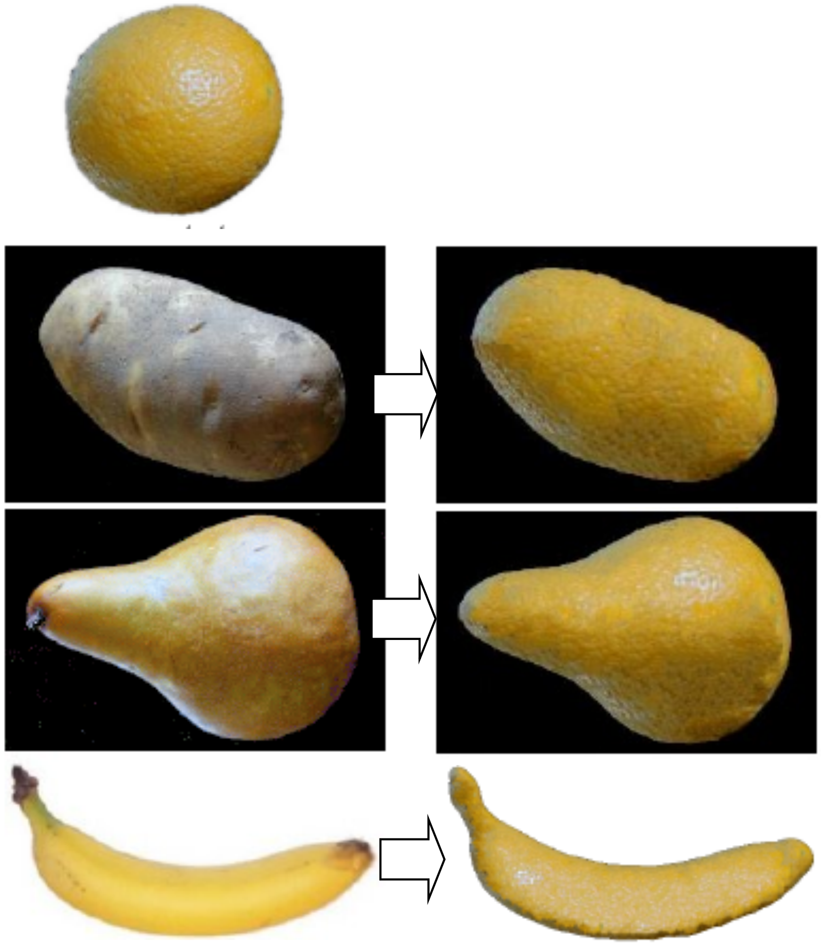
Image Quilting: Results

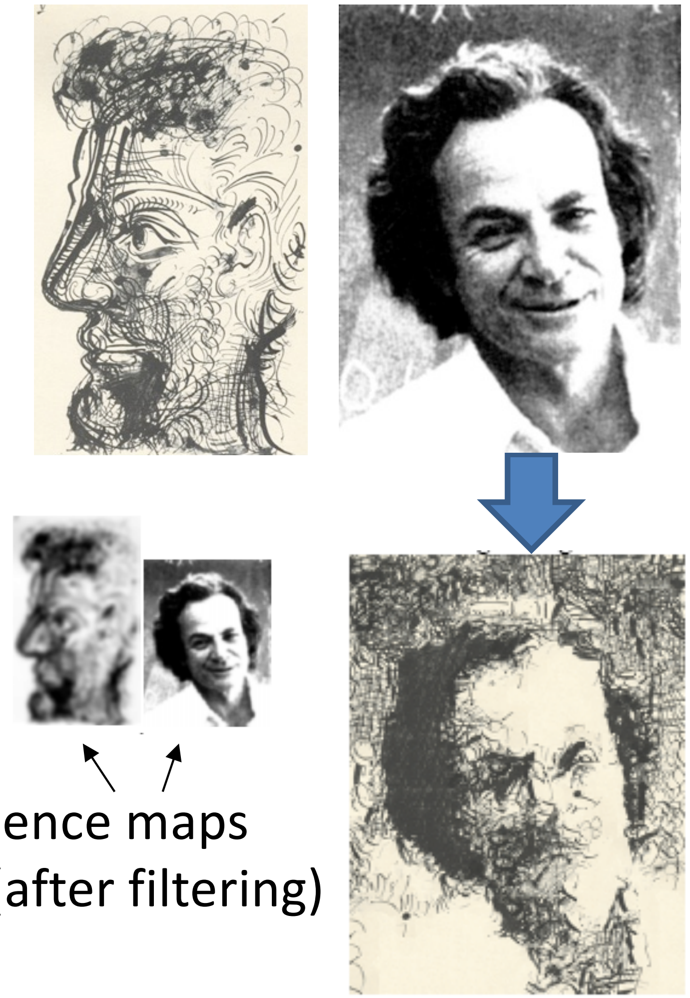




- ✓ Works rather well for all texture types
- ✗ Computationally intensive
- ✗ The influence of the block size is difficult to define

Image Quilting and texture transfer

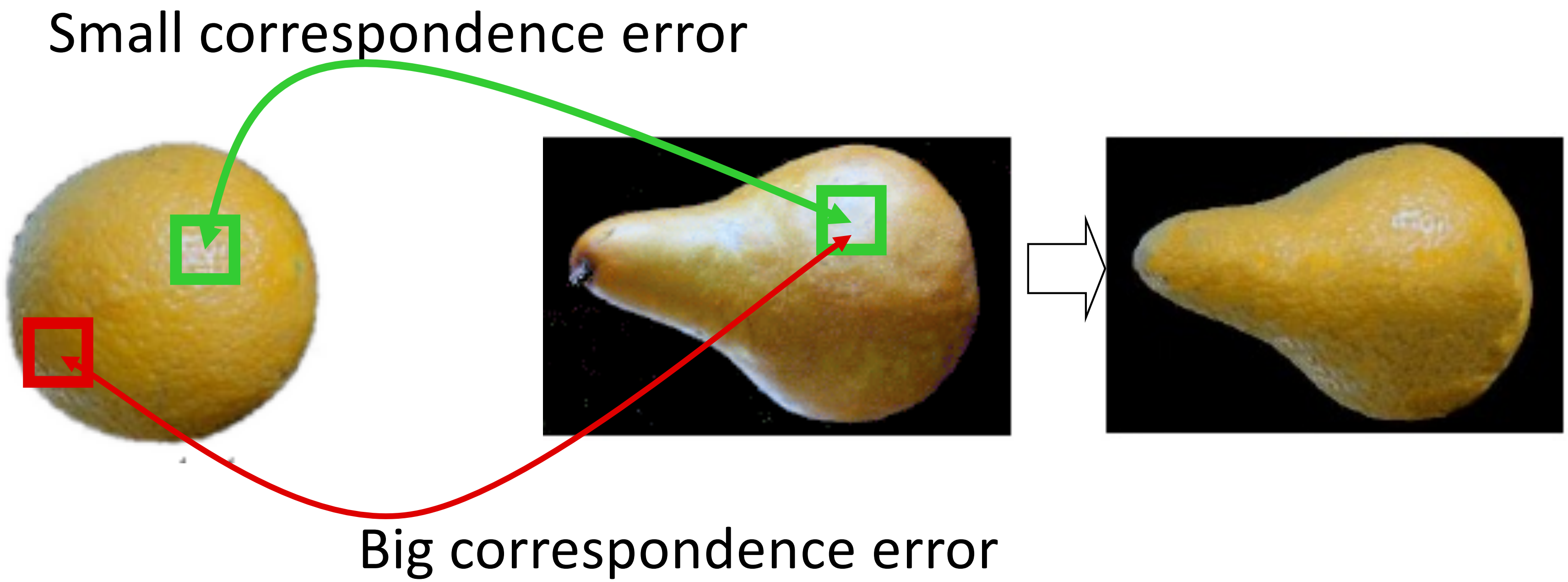




Step1: $F_{total} = \alpha * F_{block\ matching} + (1-\alpha) * F_{correspondence}$

Step2: Same as in standard "image quilting" algorithm

Image Quilting and Texture Transfer



$$F_{total} = \alpha * F_{block\ matching} + (1-\alpha) * F_{correspondence}$$

$F_{correspondence}$ measures the difference in luminance of the block from the source texture and the block from the target object

Image Quilting: Summary

- A tiling method
- “Glues” blocks from the input texture sample
- Works for almost all texture types
- Computationally intensive
- Reading:
  - A. Efros and W.T. Freeman, “Image Quilting for Texture Synthesis and Transfer,” Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp: 341 – 346, 2001.

  → All of it

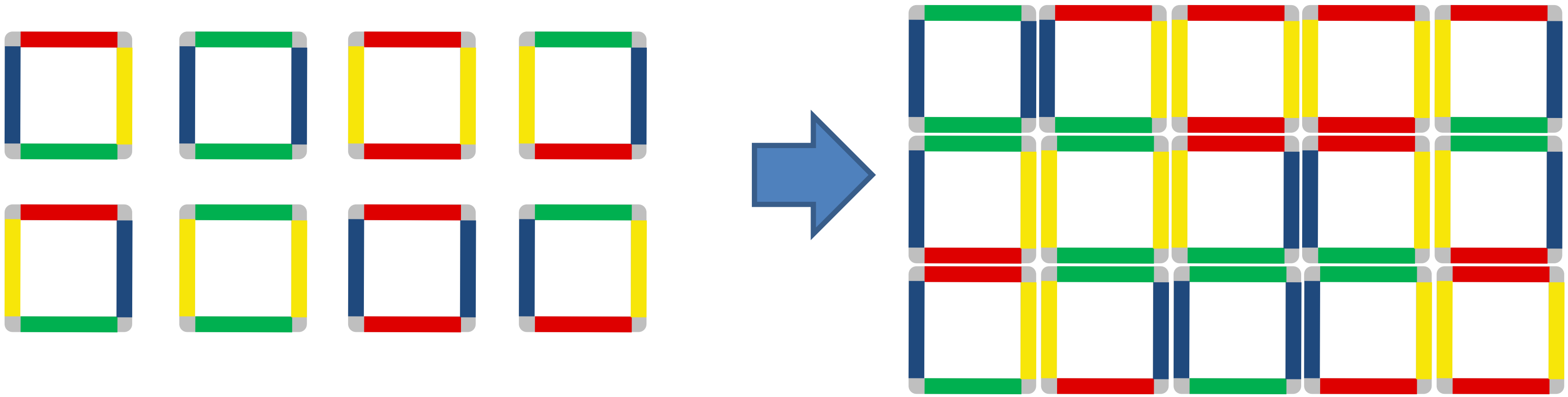
Wang Tiles

Wang Tiles

- Square tiles with color-coded edges.
- Cannot be rotated in the tiling!

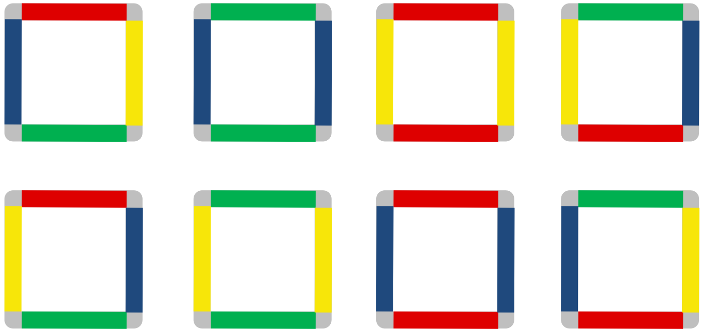
An example of a set with 2 horizontal and 2 vertical colors:

Correct tiling: adjacent edges have the same color

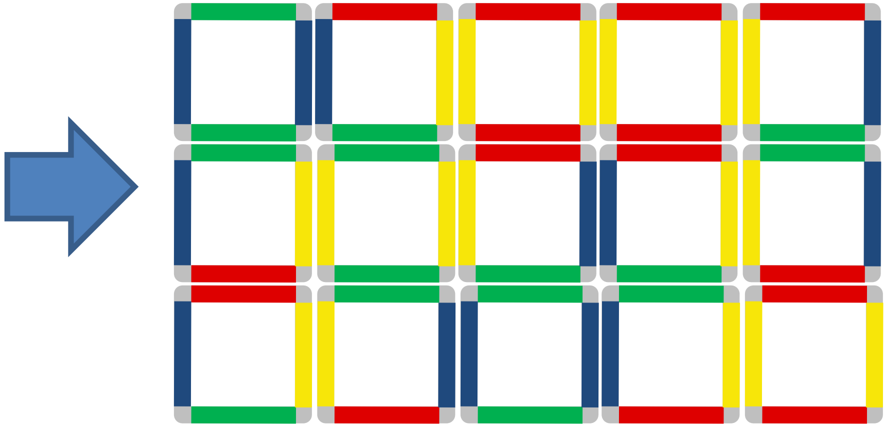


Wang Tiles

An example of a set with 2 horizontal and 2 vertical colors:

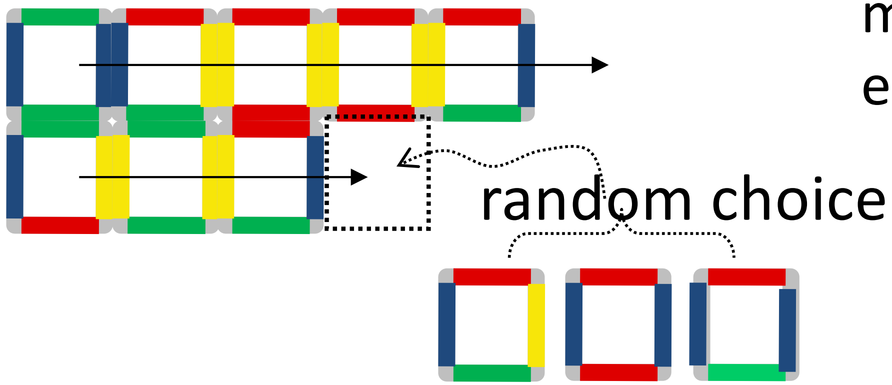


Correct tiling: adjacent edges have the same color



- This tiling was named after Hao Wang, who conjectured in 1961 that any set that can produce a valid tiling of the plane must also be able to produce a periodic tiling
  - Later showed: not true! The smallest aperiodic tiling consists of 13 tiles only.
- In the computer graphics we use sets that make periodic and not periodic tiling (e.g. the set of 8 tiles above)

Wang Tiles: Stochastic Tiling



Random choice from tiles with matching colors of North-West (NW) edges

- If there is at least one tile for each N-W combination → correct tiling is guaranteed
- If there are at least two tiles for each N-W combination → not periodic tiling is guaranteed. Why?

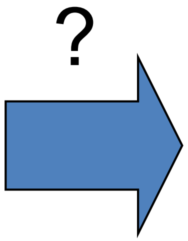
→ The smallest set that satisfies this has 8 tiles. Prove this.

(Hint: how many N-W combinations are there if there are 2 colors?)

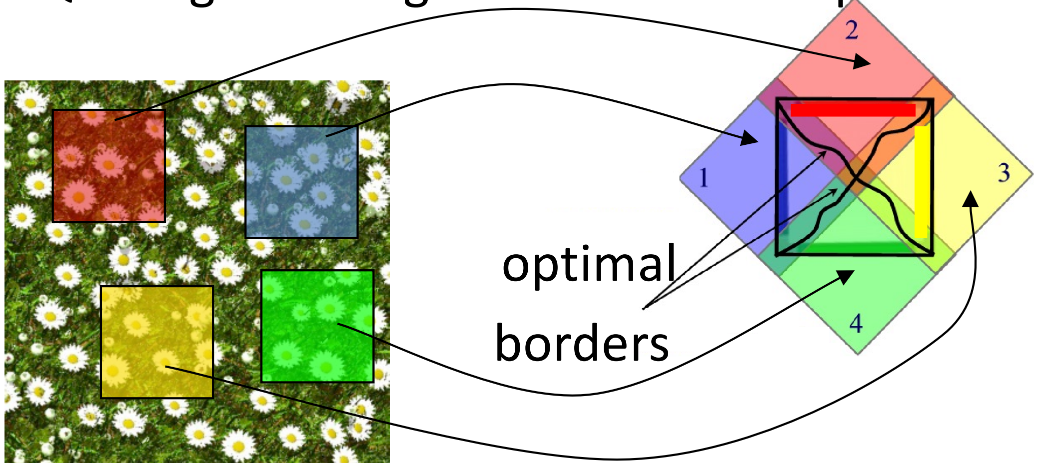
Wang Tiles: Tile Design by Quilting

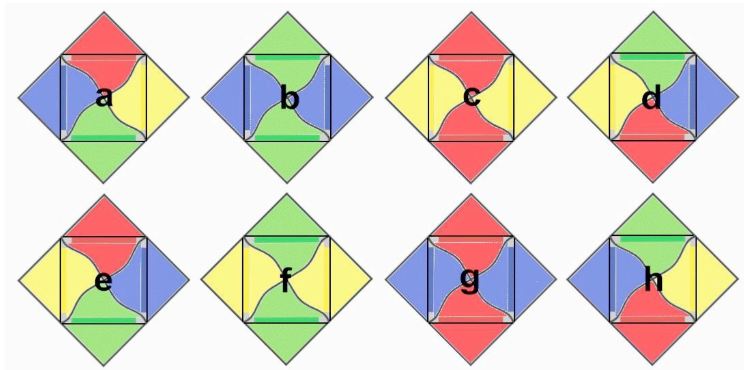
How do we achieve that the tiles perfectly match?

Example texture



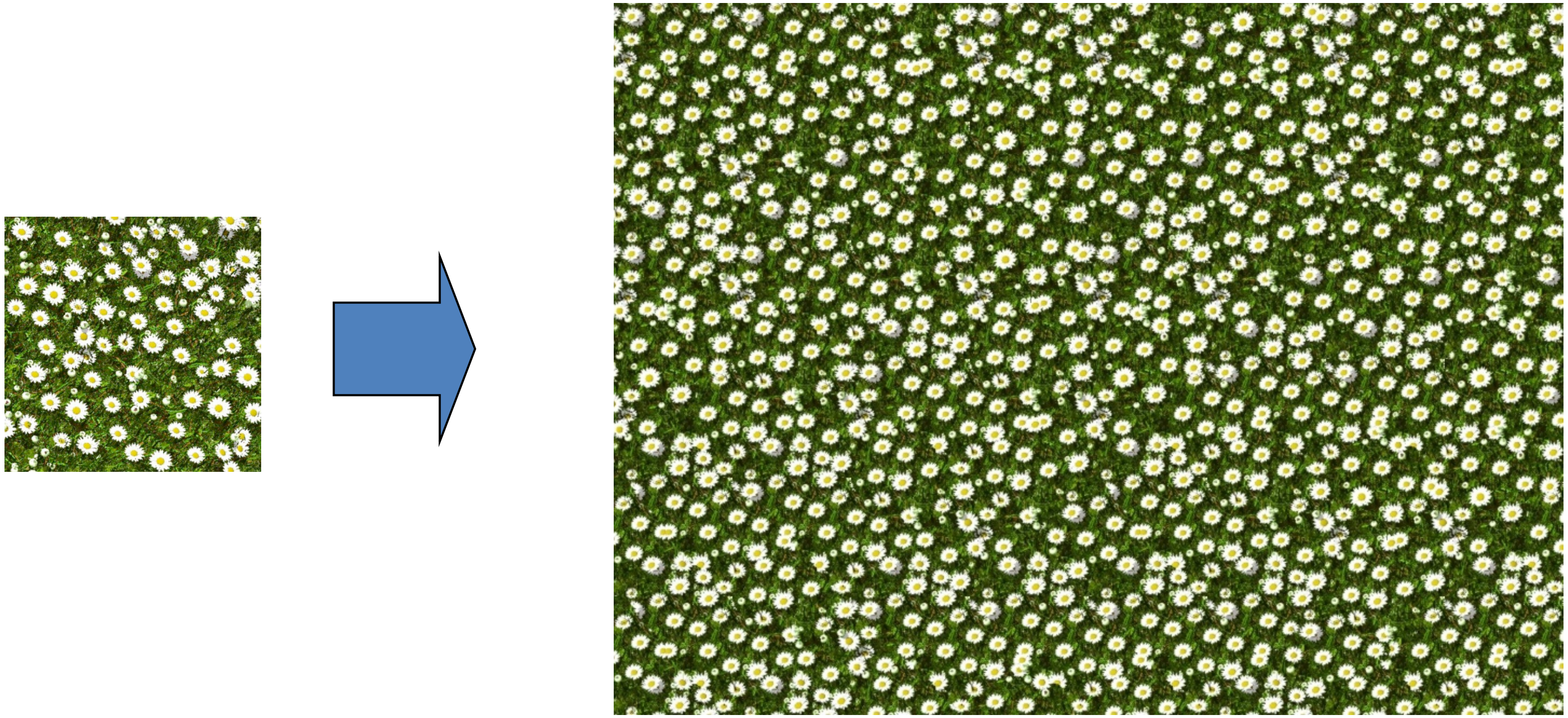
Quilting + cutting out the middle square





The remaining ones with other combinations of the same 4 blocks

Introducing Inhomogeneity

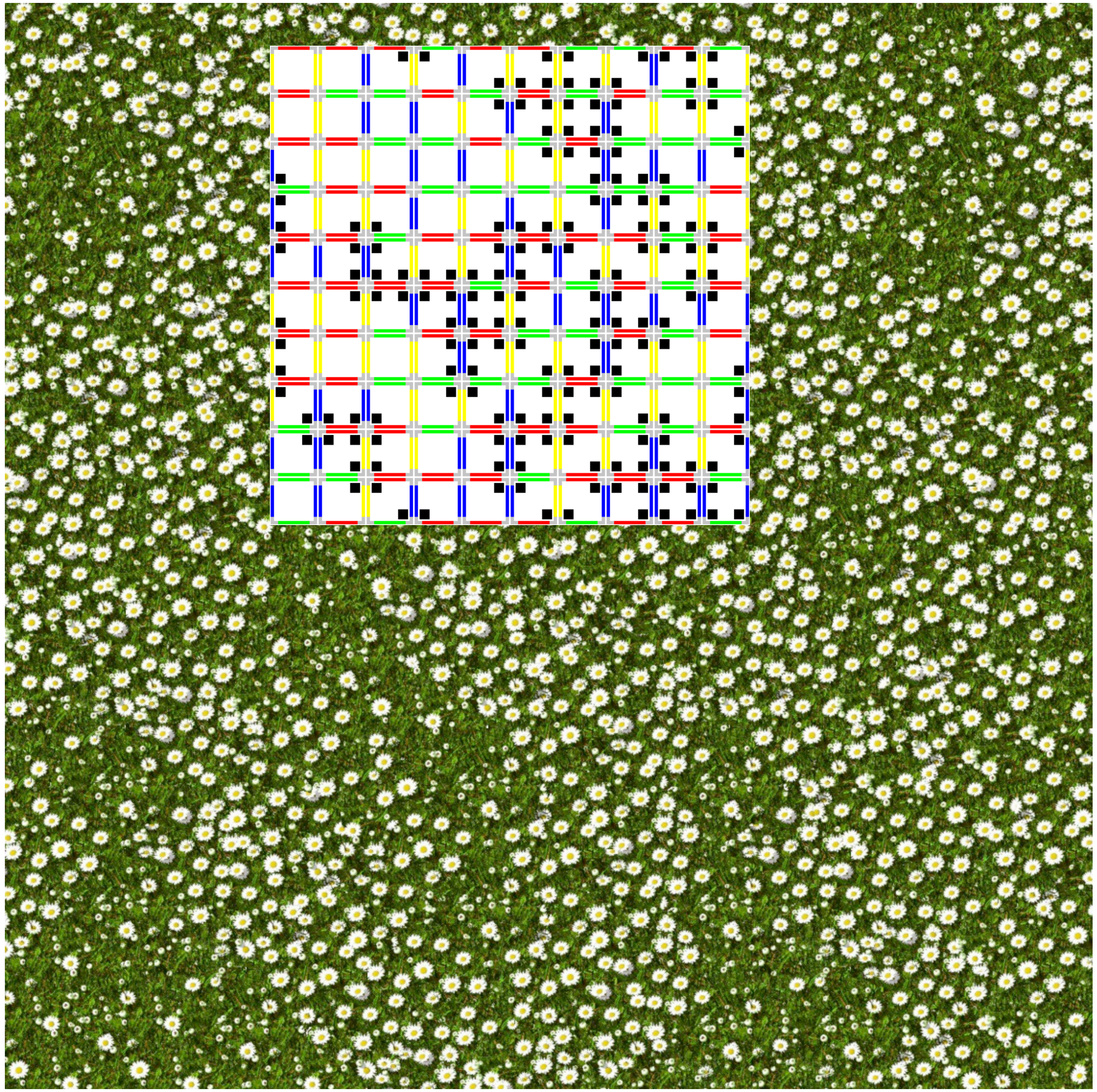


We learned to solve this: extending to arbitrary size, no copying or tiling obvious, but does it look natural?

Introducing Inhomogeneity

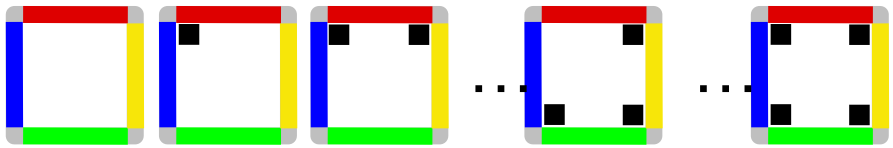
How to achieve this?

Introducing Inhomogeneity



An additional coding in each tile

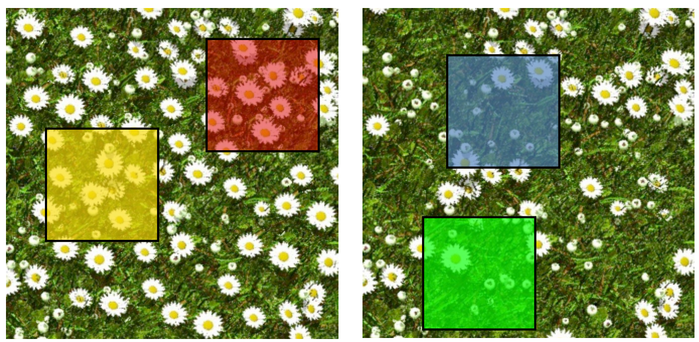
Introducing Inhomogeneity

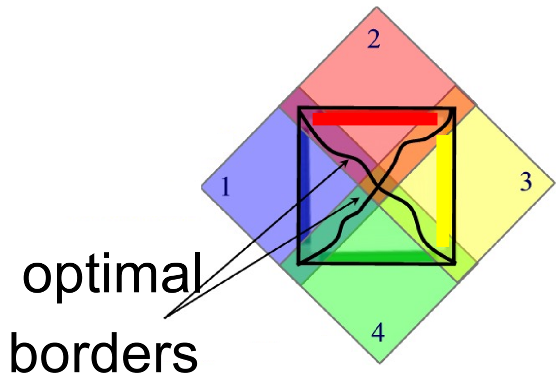


Corners are binary coded

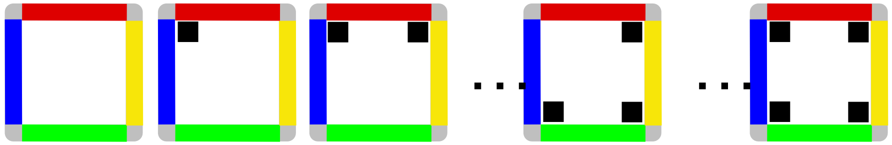
In our example: corner marked means ‘dense flowers’, corner not marked means ‘not dense flowers’

Two source textures for generating parts of tiles with and without corner marks





Introducing Inhomogeneity



Corners are binary coded

- For each tile: 2^4=16 possible corner codings
    - A set of 8 tiles extends to 8x16=128 tiles
- This can be reduced to 64 and still keeping the stochastic selection. Why? (How many corners do we need to match at each step?)

Wang Tiles: Summary

- The process of making tiles is complex (quilting)
- Once tiles are made, tiling very fast
- Works for almost all texture types
- Very powerful for producing non-homogeneous textures
- Reading:
  - Michael F. Cohen, Jonathan Shade, Stefan Hiller, Oliver Deussen, “Wang Tiles for Image and Texture Generation,” ACM Transactions on Graphics, vol. 22 , no. 3, pp. 287 – 294, July 2003.

  → Sections 1,2 and 3

Summary

- Pyramidal methods based on first-order statistics
  - Work well for stochastic textures
  - Not for non-homogeneous textures, regular patterns and complex composites
- Image Quilting
  - Conceptually simple but computationally intensive
  - For all textures (sometimes there are artifacts)
  - Very good for texture transfer and for making Wang tiles
- Wang Tiles
  - Fast and simple, making infinitely large textures
  - Very good for complex textures
  - Making not homogeneous textures: coded corners and edges