Digital Image Processing

25 January 2007

Dr. ir. Aleksandra Pizurica
Prof. Dr. Ir. Wilfried Philips

Aleksandra.Pizurica @telin.UGent.be
Tel: 09/264.3415

UNIVERSITEIT
GENT
Telecommunicatie en Informatieverwerking

INSTITUUT VOOR PERMANENTE VORMING

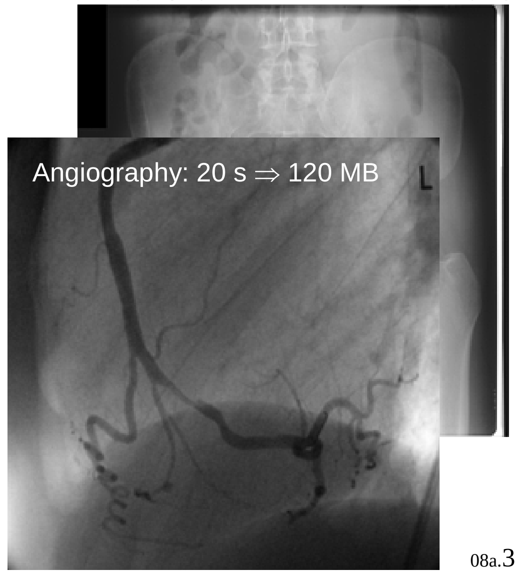
Image Compression

- Repeating -

Need for compression: Examples

"pre-press" applications

radiography 2048x1680x10 (4 Mbyte)

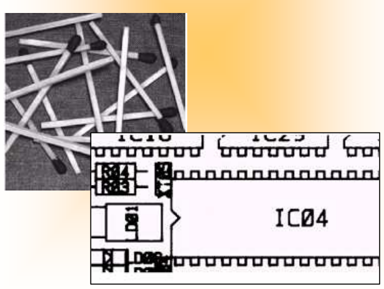


Introduction to image compression

Image compression makes use of spatial and psychovisual redundancies

Spatial (inter-pixel) redundancy

In most images, the value of any pixel can be reasonably well predicted from its neighbors. (At which positions in the image this does not hold or holds less well?)



Psychovisual redundancy

Human eye is less sensitive to certain information (e.g., very high spatial frequencies in highly textured or in dark areas). Such information is psychovisually redundant and can be removed without significantly damaging the perceptual quality

Variable length coding

	fixed length	variable length
A	000	11110
B	001	1101
C	010	101
D	011	0
E	100	100
F	101	1100
G	110	1110
H	111	11111

A D F D G D D D D C D E

000011101011110011011011011010011100

11110011000111000001010100

Principle: encode frequent symbols with a short bit string (small l_i) and rarely appearing symbols with a long bit string

Entropy

Stationary source → Symbols with possible values α_i, $i=1\ldots N$, and probability $P(\alpha_i)=p_i$

Entropy of the source $H = -\sum_{k=1}^{N} p_k \log_2 p_k$ bits (remark: $0\log_2 0=0$)

H is the smallest possible mean value of the code word length for any technique that encodes symbol per symbol (and for any coding technique if the successive symbols are statistically independent)

0 bit $\leq H \leq \log_2 N$ bit, N - number of symbols of the alphabet

H is maximal when all the symbols are equally probable: $H_{\max} = \log_2 N$

Entropy coding →
- Huffman coding } best known
- Arithmetic coding } best known
- …

Huffman coding: example…

The Huffman code is a specific pre-fix code

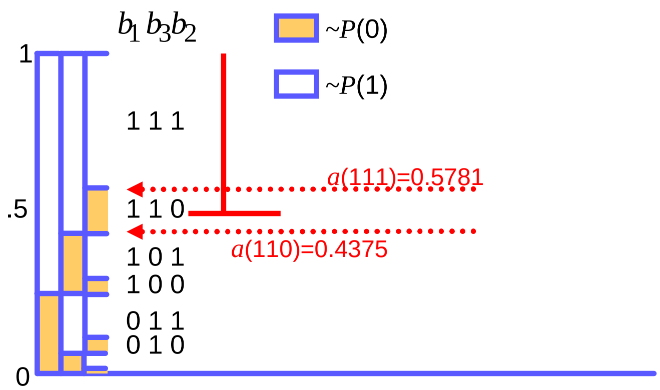
- the code word length $l_k \approx -\log_2 p_k$

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

…Huffman coding: example

Original source			Source reduction							
Sym.	Prob.	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

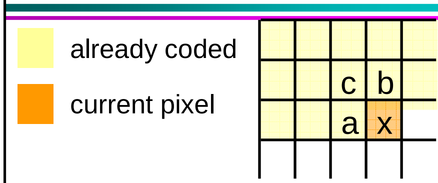
Arithmetic coding



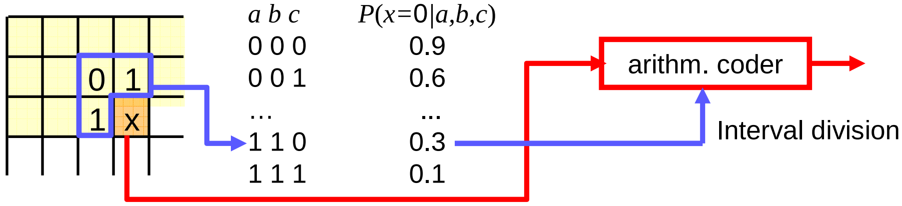
The arithmetic coder associates with every possible bit string $x_1 \ldots x_n$ a subinterval $[a(x_1 \ldots x_n), b(x_1 \ldots x_n))$ of [0,1] with the length $P(x_1 \ldots x_n)$

The code word for $x_1 \ldots x_n$ is a binary floating point representation of $a(x_1 \ldots x_n)$, but rounded up to a minimal number of bits that are needed to distinguish $a(x_1 \ldots x_n)$ from other possible $a(x'_1 \ldots x'_n)$

Context modeling: Example binary image…



Binary image: all image pixels take values 0 or 1



We code her a binary image with an arithmetic coder

The interval division of the arithmetic coder now depends on the “context”, i.e., on the values of the neighboring pixels:

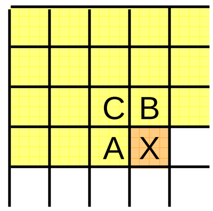
- We estimate, e.g., $P(x=0|a,b,c)$ and base on this the interval division of the coder instead of $P(x=0)$ ⇒ we make use of spatial correlation

In practice we estimate $P(x=0|a,b,c)$ adaptively for each context

Simple predictive techniques

already coded

current pixel



Prediction: $X_p = \lfloor aA + bB + cC \rfloor$

Coded error: $X\text{-}X_p$

Principle:

- The “current” pixel value is first predicted from the neighboring, previously coded ones

- The prediction error is coded using the Huffman or the arithmetic coder

Example: LJPEG (Lossless JPEG; Joint Photographic Experts Group)

The “Binary Tree Predictive Coder”...