A

Digital Image Processing (DIP): Introduction

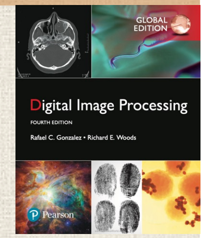
Prof. Pascal Matsakis

DIP: Introduction

I. Origins of DIP

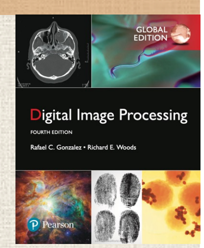
Prof. Pascal Matsakis

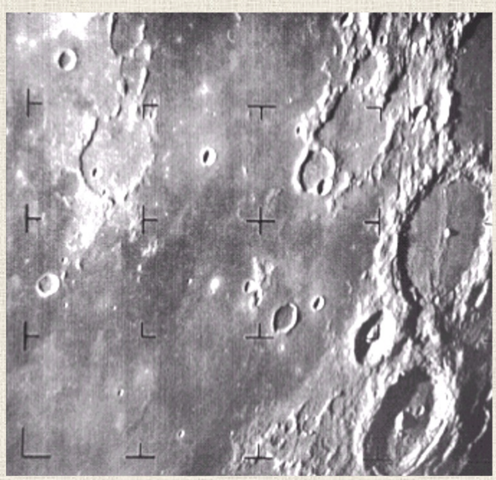
I.1. Newspaper Industry (1920s)



Digital picture produced in 1921

I.2. Space Program (1960s)





First picture of the moon
by a US spacecraft in 1964

II. Interest in DIP

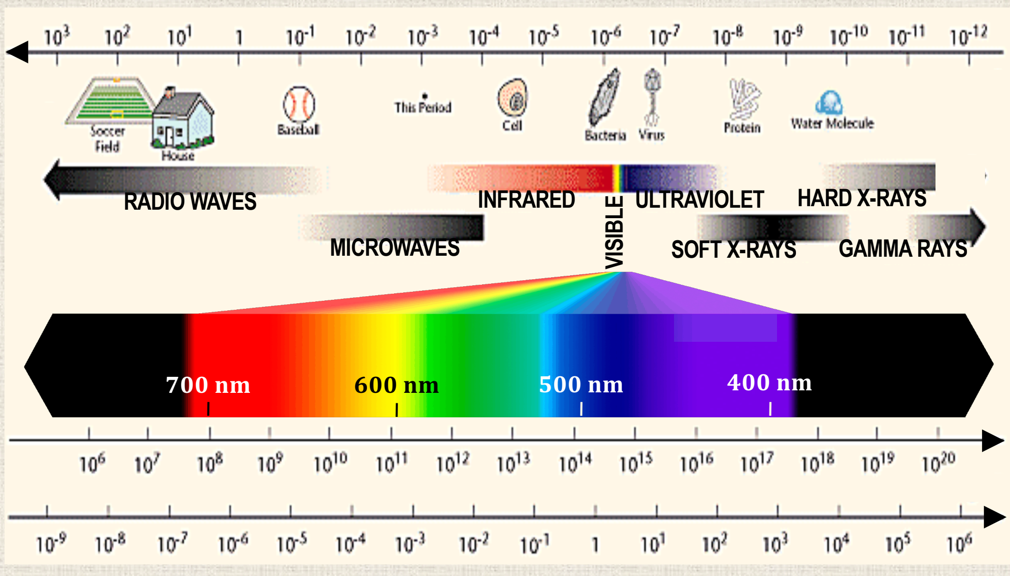
II.1. Overview

Applications in many fields:
medicine, geography, physics, astronomy, defense…

Examples:
character recognition, fingerprint recognition, target recognition, weather prediction, crop assessment, detection of bone fractures, detection of brain tumors, study of high-energy plasmas…

II.2a. Imaging in the Electromagnetic Spectrum

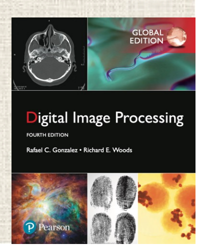
$\lambda\nu = c, \ E = h\nu$



II.2b. Imaging in the Electromagnetic Spectrum

Gamma-Ray

Bone scan
(to localize infections and tumors)



II.2c. Imaging in the Electromagnetic Spectrum

X-Ray

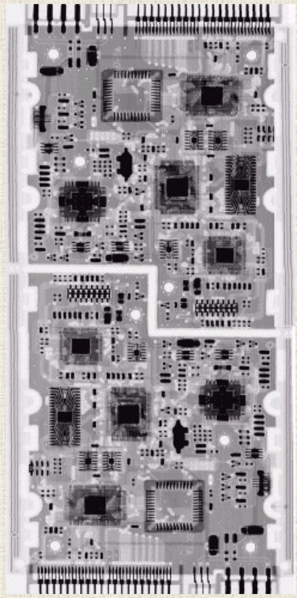
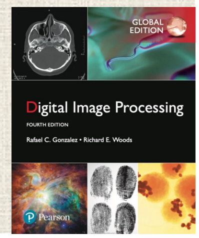
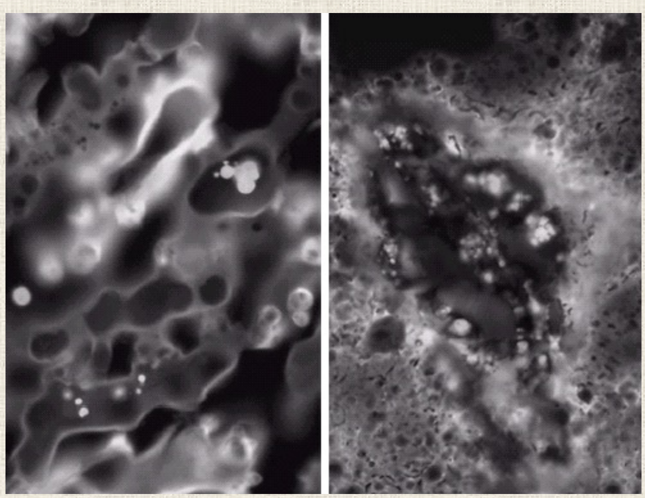


Image of a circuit board
(to detect flaws)

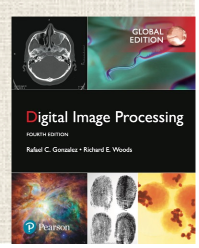


II.2d. Imaging in the Electromagnetic Spectrum

Ultraviolet



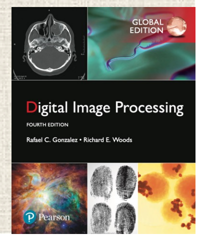
Fluorescence microscope images of corn
(to find if corn is infected by smut)



II.2e. Imaging in the Electromagnetic Spectrum

Visual Spectrum

Image of paper currency
(to track and identify bills)

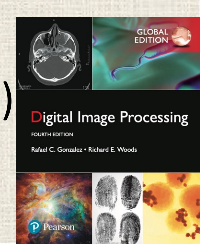


Prof. Pascal Matsakis

II.2f. Imaging in the Electromagnetic Spectrum

Infrared

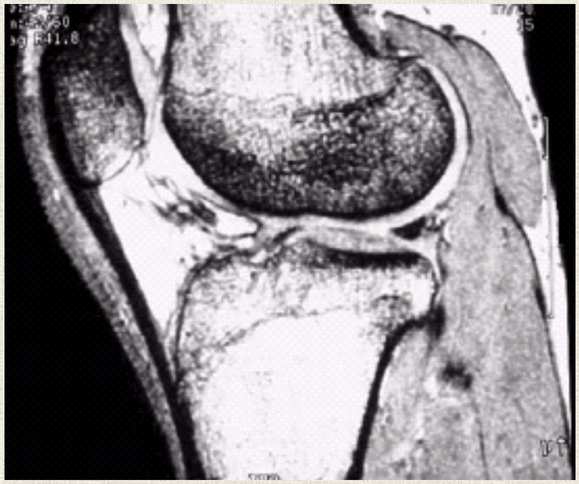
Satellite image of North America
(global inventory of human settlements)



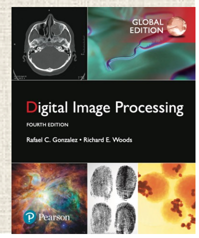
Prof. Pascal Matsakis

II.2g. Imaging in the Electromagnetic Spectrum

Radio-Band



MRI image of a human knee
(diagnosis of sport-related injuries)



Prof. Pascal Matsakis

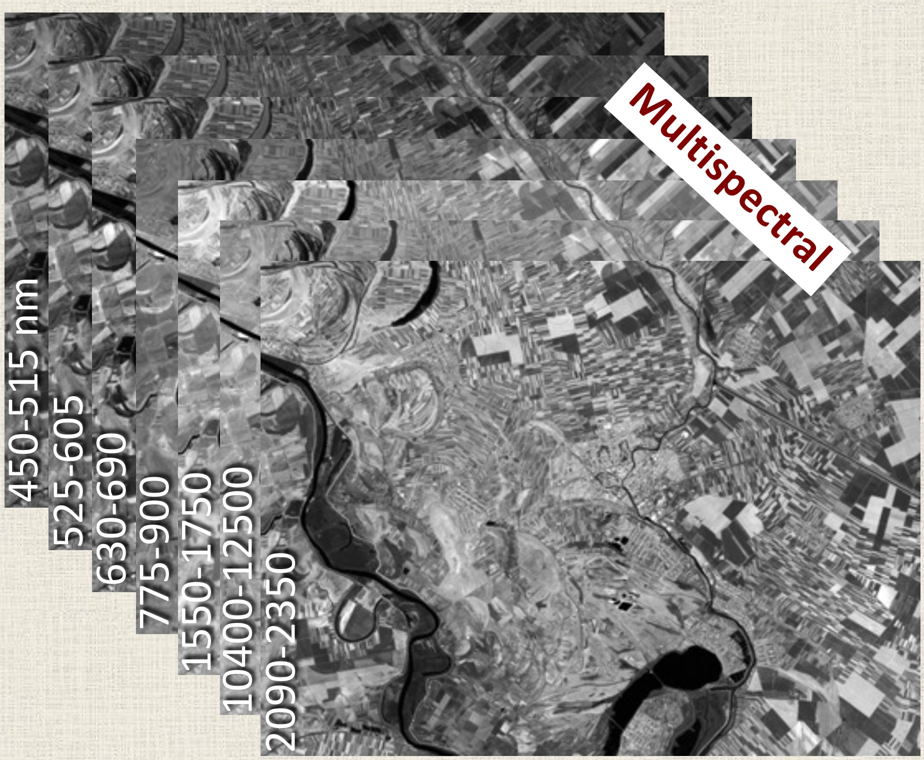
II.2h. Imaging in the Electromagnetic Spectrum

Prof. Pascal Matsakis

II.2i. Imaging in the Electromagnetic Spectrum

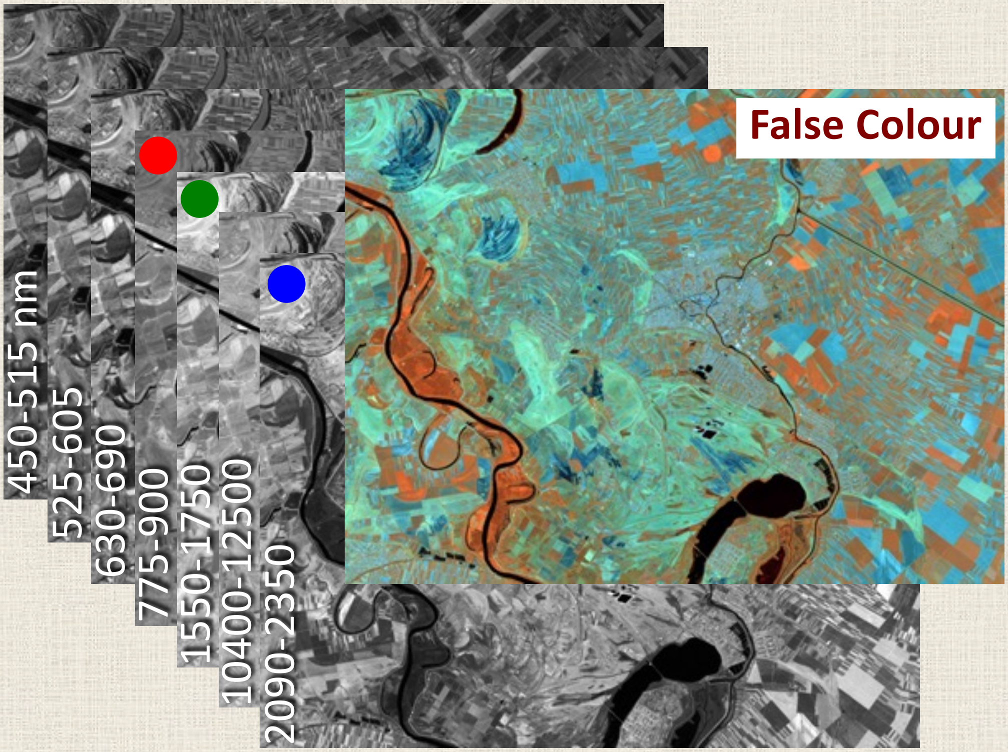
Prof. Pascal Matsakis

II.2j. Imaging in the Electromagnetic Spectrum



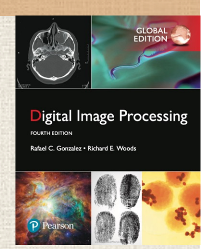
Prof. Pascal Matsakis

II.2k. Imaging in the Electromagnetic Spectrum



Prof. Pascal Matsakis

II.3a. Other Imaging Modalities

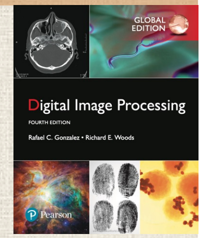


Ultrasound

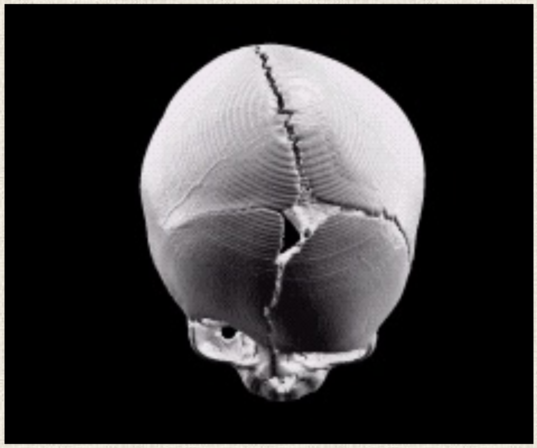
Image of an unborn baby
(to determine the health of their development)

Prof. Pascal Matsakis

II.3b. Other Imaging Modalities



Synthetic



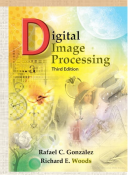
Computer-generated image of a human skull
(for criminal forensics)

B

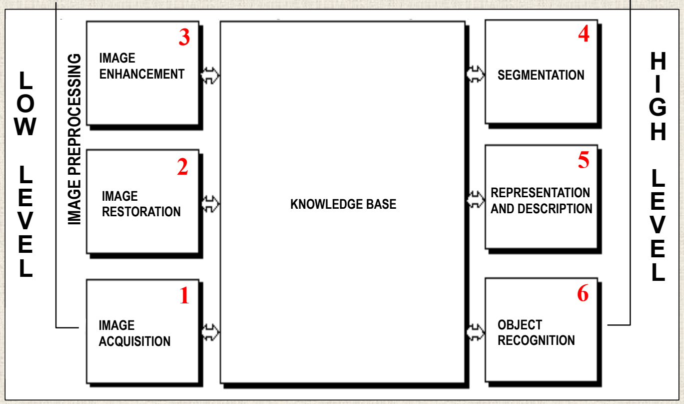
Digital Image Processing (DIP): Fundamentals

I. Main Steps in DIP

I.1a. Overview



outputs are images outputs are image attributes



I.1b. Overview

LOW LEVEL:
acquisition and improvement of pictorial information for human interpretation and analysis

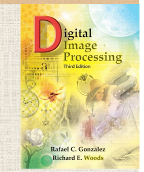
HIGH LEVEL:
autonomous machine perception

I.1c. Overview

- DIP characterized by specific solutions.
- Techniques that work well in one area can be totally inadequate in another.
- Actual solution of a specific problem generally still requires significant research and development.

Prof. Pascal Matsakis

I.2a. Examples



Which step?

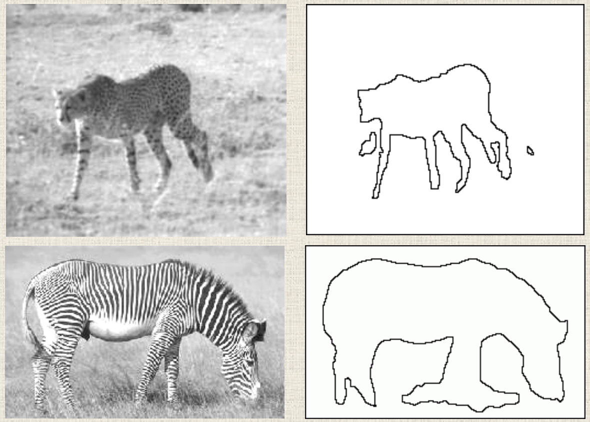
Prof. Pascal Matsakis

I.2b. Examples

Which step?

Prof. Pascal Matsakis

I.2c. Examples



Which step?

Prof. Pascal Matsakis

I.2d. Examples

Which step?

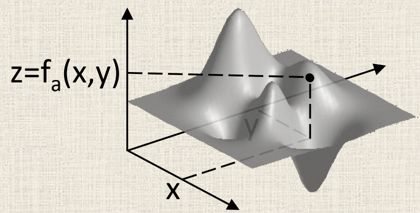
II. Image Definition and Representation

II.1a. Analog Image

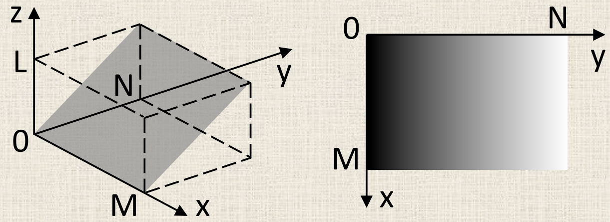
image
- function $f_a \mid \mathbb{R}^2 \to [0;+\infty[$
- function $f_a \mid \mathbb{R}^3 \to \mathbb{R}$
- function $f_a \mid \mathbb{R}^2 \to \mathbb{R}^3$
- function $f_a \mid \mathbb{R}^2 \to \mathbb{C}$

$f_a \mid \mathbb{R}^2 \to [0;+\infty[$ is a typical case;
$f_a(x,y)$ is the **intensity** of f_a at (x,y).

II.1b. Analog Image



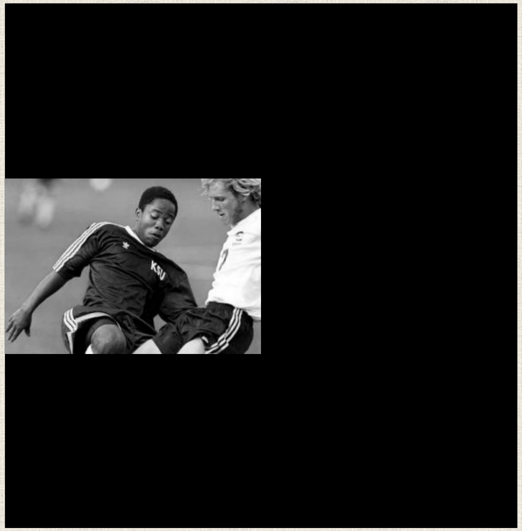
Consider $f_a \mid \mathbb{R}^2 \to [0;+\infty[$



Domain of definition [0,M]x[0,N] and range [0,L]

II.1c. Analog Image

f_a



Zero Padding

Prof. Pascal Matsakis

II.1d. Analog Image

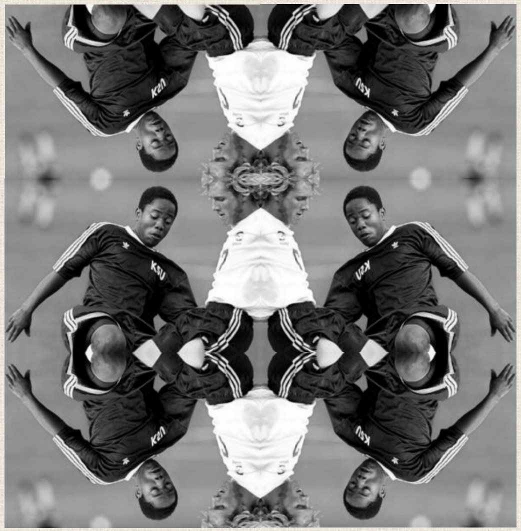
f_a

Circular Indexing

Prof. Pascal Matsakis

II.1e. Analog Image

f_a



Reflected Indexing

Prof. Pascal Matsakis

II.2a. Digital Image

image
- function $f \mid \mathbb{Z}^2 \to 0..+\infty$
- function $f \mid \mathbb{Z}^3 \to \mathbb{Z}$
- function $f \mid \mathbb{Z}^2 \to \mathbb{Z}^3$

$f \mid \mathbb{Z}^2 \to 0..+\infty$ is a typical case: **grayscale image**

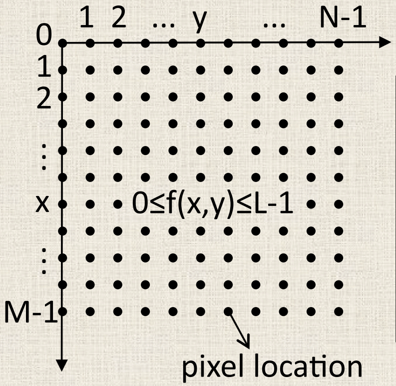
((x,y),f(x,y))

location **gray level** **pixel picture element**

Prof. Pascal Matsakis

II.2b. Digital Image

Consider $f \mid \mathbb{Z}^2 \rightarrow 0..+\infty$
Assume range included in 0..L-1:
typically, $L=2^\ell$, i.e., **ℓ-bit grayscale image**
Assume domain of definition is 0..M-1 x 0..N-1:

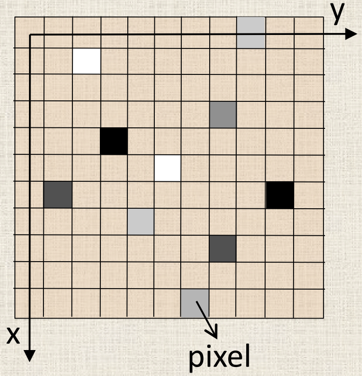


$$\begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & & & \\ \vdots & & & \vdots \\ f(M-1,0) & & \dots & f(M-1,N-1) \end{bmatrix}$$

Prof. Pascal Matsakis

II.2c. Digital Image

Consider $f \mid \mathbb{Z}^2 \rightarrow 0..+\infty$
Assume range included in 0..L-1:
typically, $L=2^\ell$, i.e., **ℓ-bit grayscale image**
Assume domain of definition is 0..M-1 x 0..N-1:



$$\begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & & & \\ \vdots & & & \vdots \\ f(M-1,0) & & \dots & f(M-1,N-1) \end{bmatrix}$$

Prof. Pascal Matsakis

II.3a. Digitization

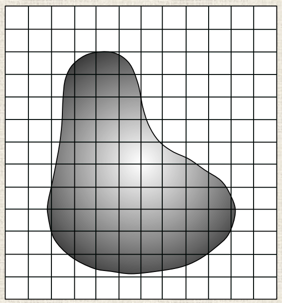
$f_a \mid \mathbb{R}^2 \rightarrow [0;+\infty[$

digitization ↓ **sampling** ↓ **quantization** ↓

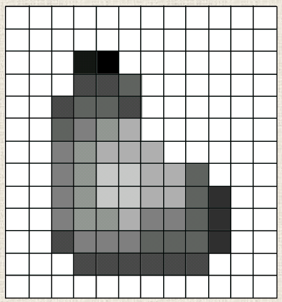
$\mathbb{Z}^2 \rightarrow [0;+\infty[$
$\mathbb{R}^2 \rightarrow 0..+\infty$

$f \mid \mathbb{Z}^2 \rightarrow 0..+\infty$

analog

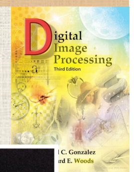


digital

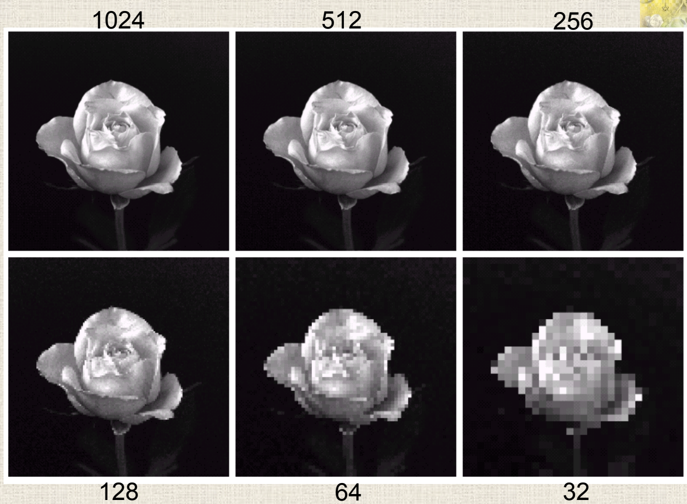


Prof. Pascal Matsakis

II.3b. Digitization

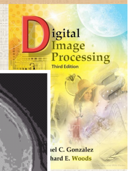


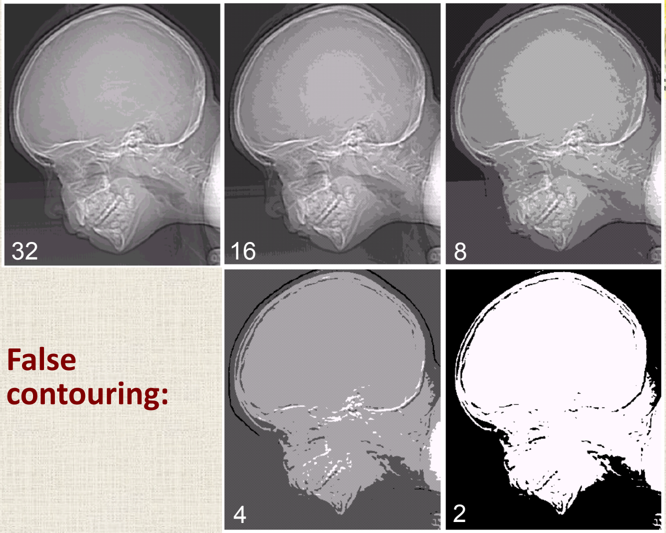
Checkerboard effect:



Prof. Pascal Matsakis

II.3c. Digitization

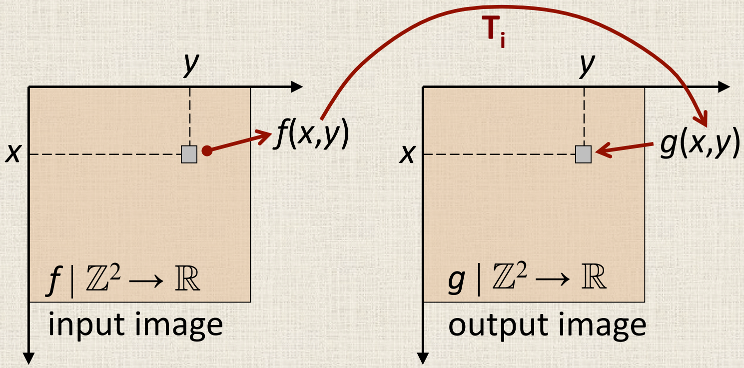




False contouring:

III. Image Operations and Transformations

III.1a. Single-Pixel Operations



$$g(x,y) = \mathrm{T_i}(f(x,y))$$

III.1b. Single-Pixel Operations

Example: $g=af+b \;\Big|\; \begin{array}{l} \mathbb{Z}^2 \longrightarrow \mathbb{R} \\ (x,y) \mapsto (af+b)(x,y) = af(x,y)+b \end{array}$

8-bit grayscale image f

g=255–f

III.1c. Single-Pixel Operations

Example: $g = af + b \; \Big| \; \begin{array}{l} \mathbb{Z}^2 \longrightarrow \mathbb{R} \\ (x,y) \mapsto (af+b)(x,y) = af(x,y)+b \end{array}$

8-bit grayscale image f

h derived from g=f+100

Prof. Pascal Matsakis

III.1d. Single-Pixel Operations

Example: $\exists (x,y) \in \mathbb{Z}^2$, $g(x,y) \notin 0..255$
"Viewable" version, h, of g?

- first possibility

$\forall (x,y) \in \mathbb{Z}^2$, $h(x,y) = \mathrm{nint}[\min\{255, \max\{0, g(x,y)\}\}]$

- second possibility

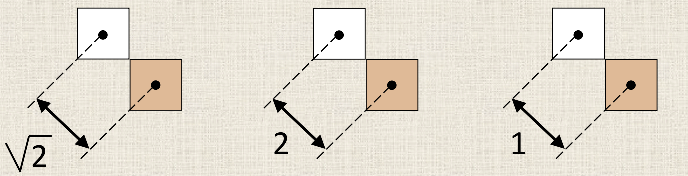
$m = \min_{(x,y) \in \mathbb{Z}^2} g(x,y)$ and $M = \max_{(x,y) \in \mathbb{Z}^2} g(x,y)$

$\forall (x,y) \in \mathbb{Z}^2$, $h(x,y) = \mathrm{nint}[255(g(x,y) - m)/(M - m)]$

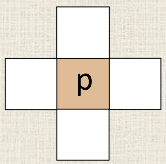
- ......

Prof. Pascal Matsakis

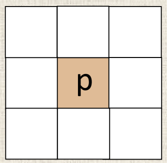
III.2a. Neighborhood Operations



Euclidean d_E | **City-block** d_4 | **Chessboard** d_8



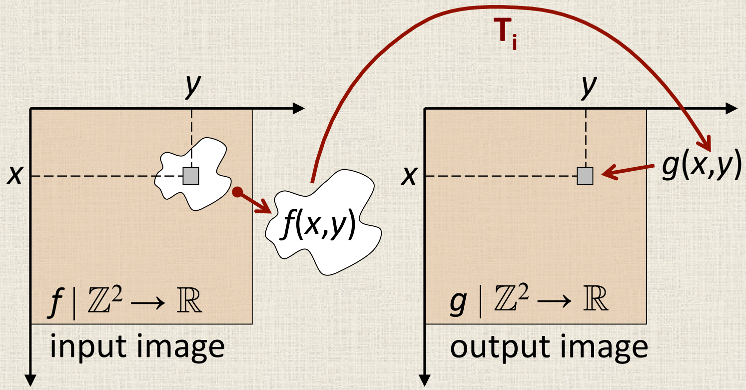
4-neighbors of p



8-neighbors of p

Prof. Pascal Matsakis

III.2b. Neighborhood Operations



$g(x,y)$ depends on the gray levels from some neighborhood of (x,y) in f.

Prof. Pascal Matsakis

III.2c. Neighborhood Operations

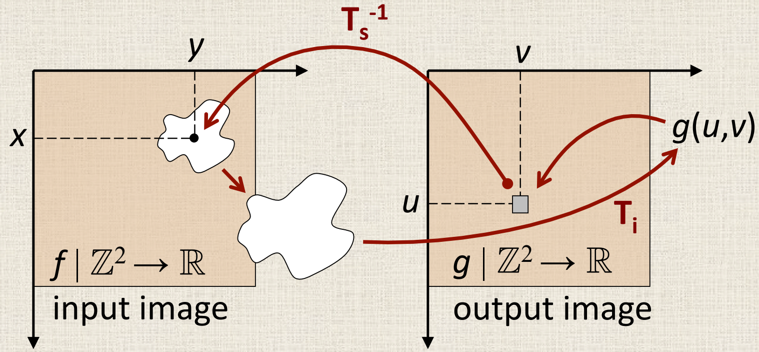
Example: $g(x,y)$ is the average of the gray levels from some neighborhood of (x,y) in f.

f

g

Prof. Pascal Matsakis

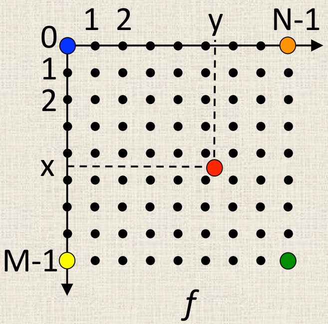
III.3a. Geometric Spatial Transformations



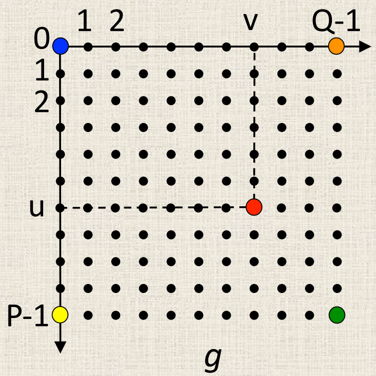
$g(u,v)$ depends on the gray levels from some neighborhood of (x,y) in f.

Prof. Pascal Matsakis

III.3b. Geometric Spatial Transformations



f



g

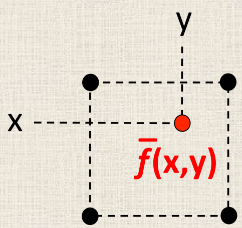
$g(0,0)=\boldsymbol{f(0,0)}$
$g(0,Q\text{-}1)=\boldsymbol{f(0,N\text{-}1)}$
$g(P\text{-}1,0)=\boldsymbol{f(M\text{-}1,0)}$
$g(P\text{-}1,Q\text{-}1)=\boldsymbol{f(M\text{-}1,N\text{-}1)}$

$g(u,v)=\boldsymbol{\bar{f}(x,y)}$
with x=? and y=?

Example: Image scaling

Prof. Pascal Matsakis

III.3c. Geometric Spatial Transformations



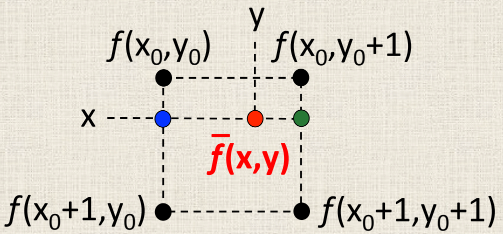
Nearest neighbour:

$\boldsymbol{\bar{f}(x,y)}=f(X,Y)$
with X=? and Y=?

Example: Image scaling

Prof. Pascal Matsakis

III.3d. Geometric Spatial Transformations



Bilinear interpolation:

$\bar{f}(x,y_0)=(1-s).f(x_0,y_0)+s.f(x_0+1,y_0)$

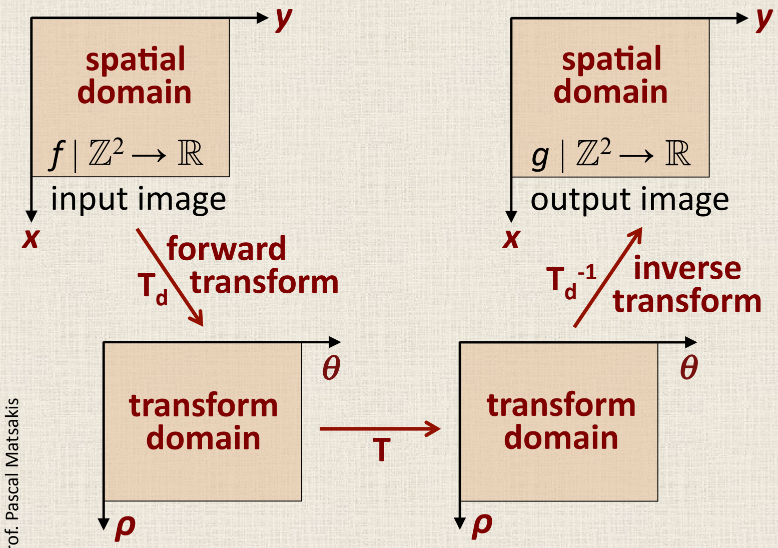
$\bar{f}(x,y_0+1)=(1-s).f(x_0,y_0+1)+s.f(x_0+1,y_0+1)$

$\bar{f}(x,y) = (1-t).\bar{f}(x,y_0)+t.\bar{f}(x,y_0+1)$

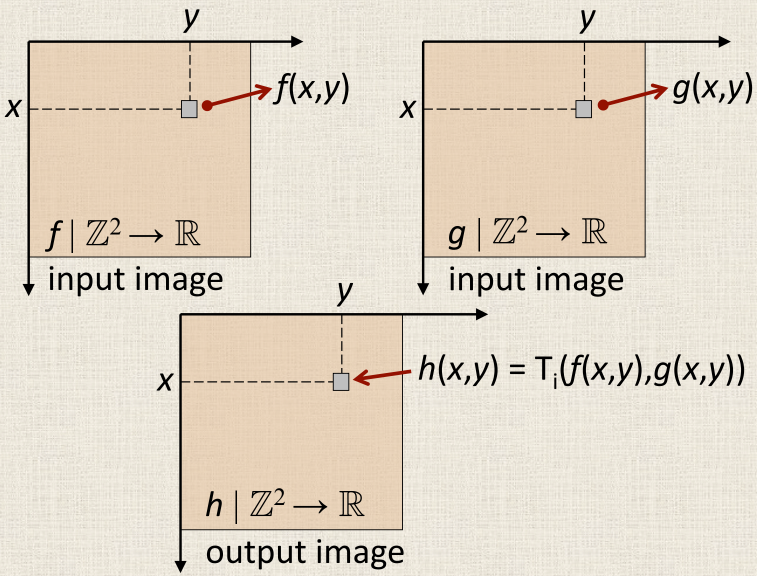
with s=? and t=?

Example: Image scaling

III.4. Image Transforms



III.5a. Binary Single-Pixel Operations



III.5b. Binary Single-Pixel Operations

Arithmetic:

$$h=f+g \left| \begin{array}{l} \mathbb{Z}^2 \to \mathbb{R} \\ (x,y) \mapsto (f+g)(x,y) = f(x,y)+g(x,y) \end{array} \right.$$

$$h=fg \left| \begin{array}{l} \mathbb{Z}^2 \to \mathbb{R} \\ (x,y) \mapsto (fg)(x,y) = f(x,y)g(x,y) \end{array} \right.$$

III.5c. Binary Single-Pixel Operations

f

g

h=|f−g|

Prof. Pascal Matsakis

III.5d. Binary Single-Pixel Operations

Logical:

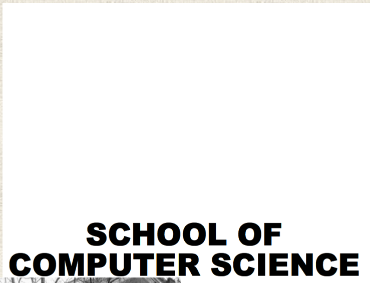
$$h=f_{\wedge}g \;\Big|\; \begin{array}{l} \mathbb{Z}^2 \rightarrow \mathbb{R} \\ (x,y) \mapsto (f \wedge g)(x,y) = \min\{f(x,y),g(x,y)\} \end{array}$$

$$h=f_{\vee}g \;\Big|\; \begin{array}{l} \mathbb{Z}^2 \rightarrow \mathbb{R} \\ (x,y) \mapsto (f \vee g)(x,y) = \max\{f(x,y),g(x,y)\} \end{array}$$

Prof. Pascal Matsakis

III.5e. Binary Single-Pixel Operations

f

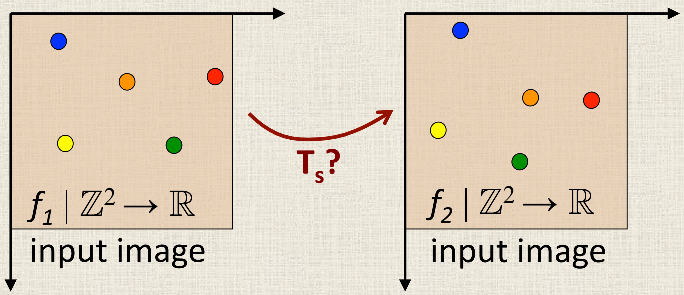


g

$f \vee \neg g$

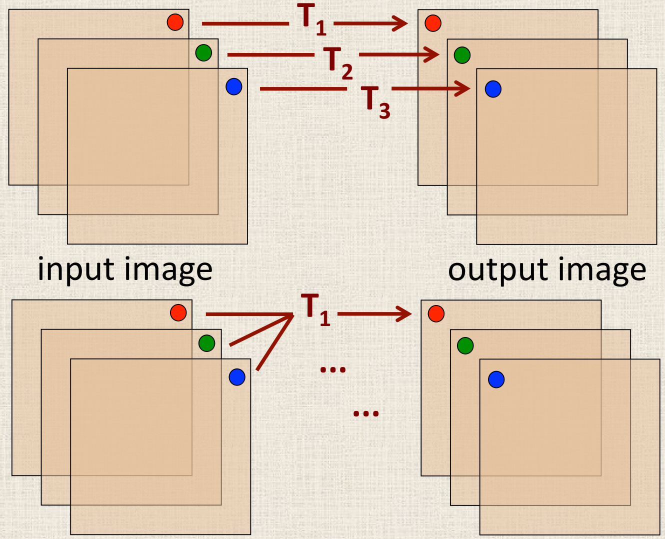
Prof. Pascal Matsakis

III.6. Image Registration



Prof. Pascal Matsakis

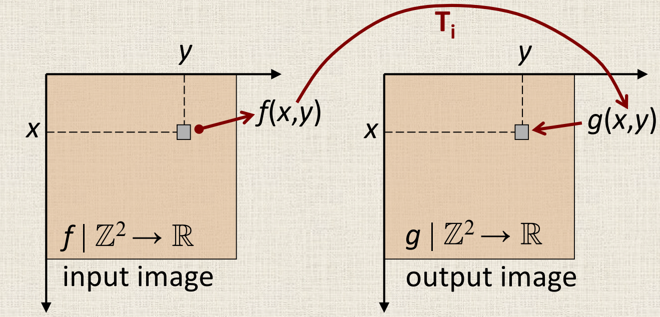
III.7. Colour/Multispectral Transformations



C

Single-Pixel Operations

Prof. Pascal Matsakis



$$g(x,y) = T_i(f(x,y))$$

Prof. Pascal Matsakis

Single-Pixel Operations

I. Gray-Level Mappings

Prof. Pascal Matsakis

I.1. What Are They?

Digital image **f**, number of gray-level values L
Gray-level mapping **m** | $\mathbb{R} \to \mathbb{R}$

→ Digital image **g**, number of gray-level values L

$g(x,y) = m(\cancel{f(x,y)})$
$= \textit{nint}\ (\ \textit{max}\ \{ 0 , \textit{min}\ \{L-1, \mathbf{m(f(x,y))}\} \}\ )$

m(u)=u+2

f → g

f:

0	3	1	1
1	2	6	0
5	3	2	2
0	2	0	3

L=8

g:

2	5	3	3
3	4	7	2
7	5	4	4
2	4	2	5

Prof. Pascal Matsakis

I.2a. Linear Mappings

m(u) = **a**u+**b**

b is the ***bias*** → impact on ***brightness***
- consider m(u)=u+b
- consider cases b<0 and b>0

a is the ***gain*** → impact on ***contrast***
- consider m(u)=au
- consider cases a<1 and a>1

$f \xrightarrow{m(u)=2u} g$

f:

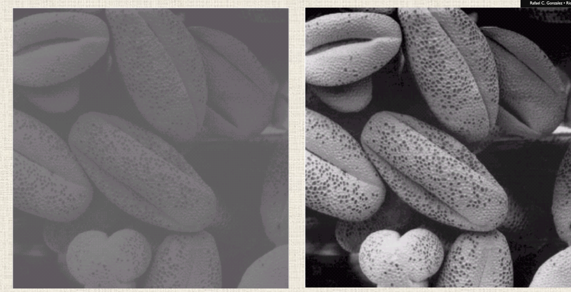
0	3	1	1
1	2	3	0
1	3	2	2
0	2	0	3

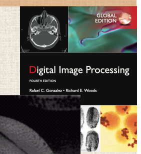
L=8

g:

0	6	2	2
2	4	6	0
2	6	4	4
0	4	0	6

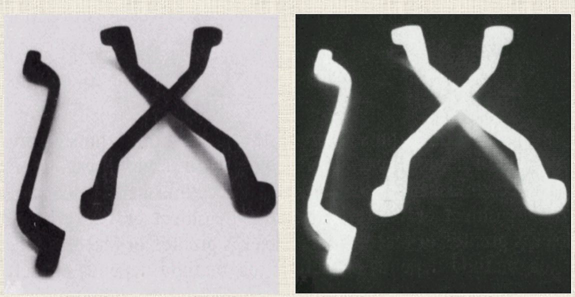
I.2b. Linear Mappings





example 1
a? b?

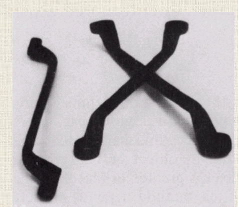
I.2c. Linear Mappings



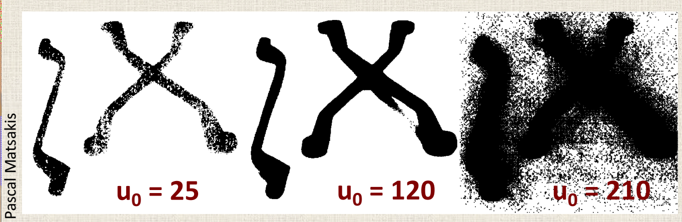
example 2
a? b?

(a = -1, b = L - 1)

I.2d. Linear Mappings

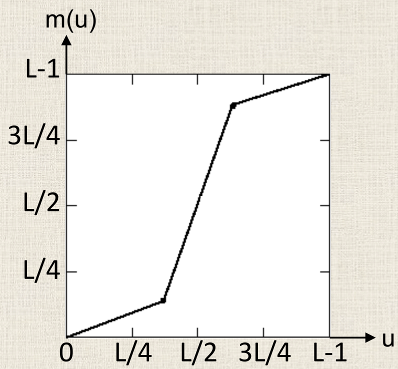


example 3
a? b?



Gray-Level Mappings 9

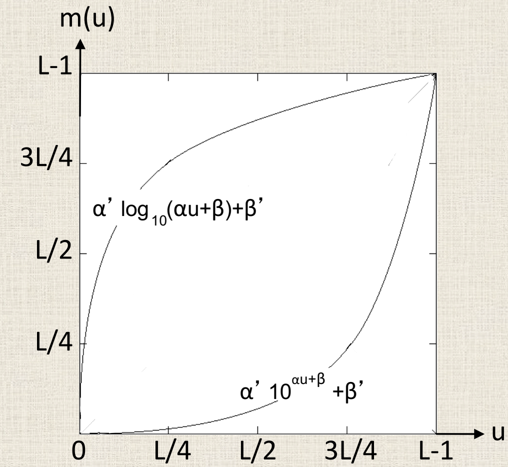
I.3. Piecewise Linear Mappings



Prof. Pascal Matsakis

Gray-Level Mappings 10

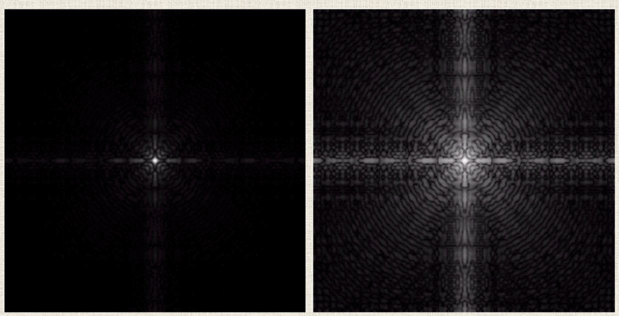
I.4a. Logarithmic and Exponential Mappings



Prof. Pascal Matsakis

Gray-Level Mappings 11

I.4b. Logarithmic and Exponential Mappings



example

m(u) = ?

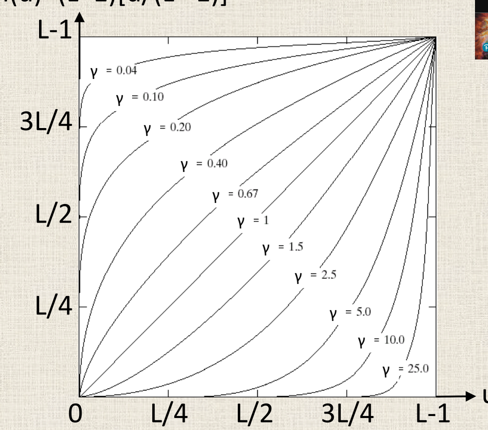
m(u) =

Prof. Pascal Matsakis

Gray-Level Mappings 12

I.5a. Power-Law Mappings

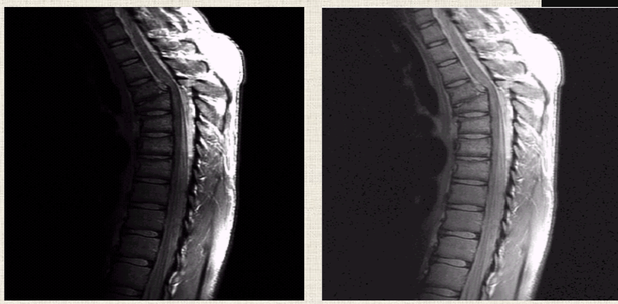
$m(u)=(L-1)[u/(L-1)]^{\gamma}$



Prof. Pascal Matsakis

Gray-Level Mappings 13

I.5b. Power-Law Mappings



example 1

γ ?

$\gamma = < 1$

Prof. Pascal Matsakis

Gray-Level Mappings 14

I.5c. Power-Law Mappings

example 2

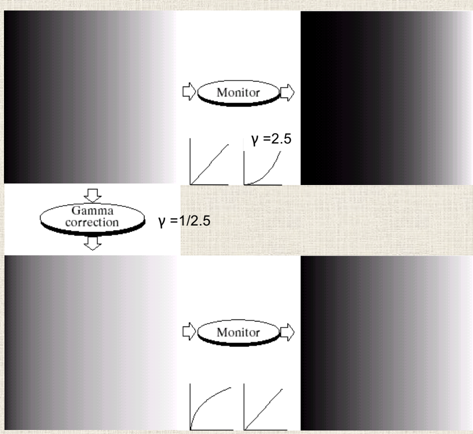
γ ?

$\gamma = > 1$

Prof. Pascal Matsakis

Gray-Level Mappings 15

I.5d. Power-Law Mappings



as viewed on monitor

as viewed on monitor

example 3

Prof. Pascal Matsakis

Gray-Level Mappings 16

I.6. Final Remarks

- Gray-level mappings useful for image enhancement (brightness and contrast adjustment)
- There is often loss of information (even if there is perceptual improvement)

$m(u)=u+2$

0	3	1	1
1	2	6	0
5	3	2	2
0	2	0	3

L=8

2	5	3	3
3	4	7	2
7	5	4	4
2	4	2	5

?

Prof. Pascal Matsakis

Single-Pixel Operations

II. Image Histograms

II.1. What Are They?

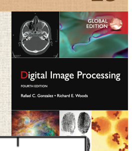
H_f histogram

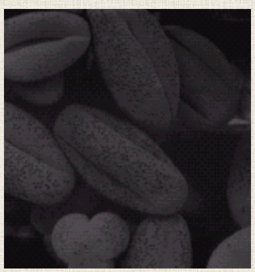
H_f^n ***normalized*** histogram

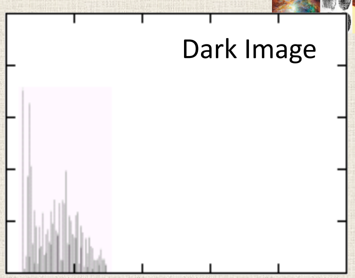
H_f^c ***cumulative*** histogram

H_f^{cn} ***cumulative normalized*** histogram

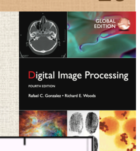
II.2a. Examples

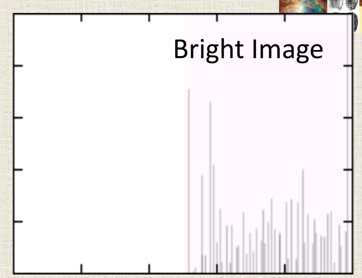




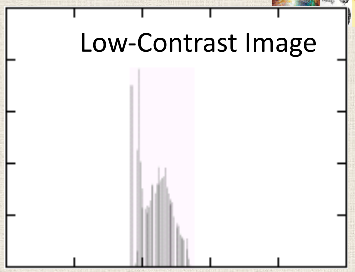


II.2b. Examples

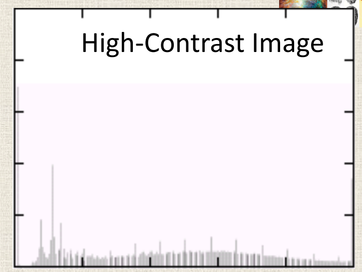




II.2c. Examples



II.2d. Examples



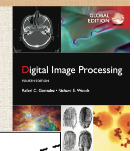
II.3. What Are They for?

- They provide a global description of the appearance of an image (brightness and contrast)
- They give us useful information for image enhancement (brightness and contrast adjustment)

II.4a. Histogram Equalization

Gray-level mapping that yields an “optimal” improvement in contrast

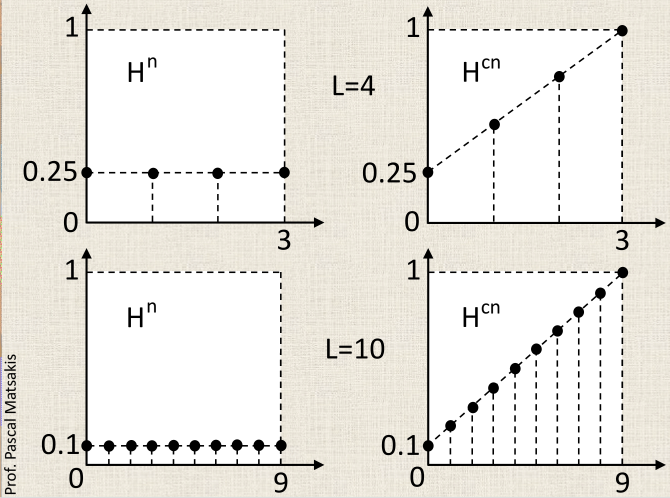
II.4b. Histogram Equalization



The mapping is specific to the image considered

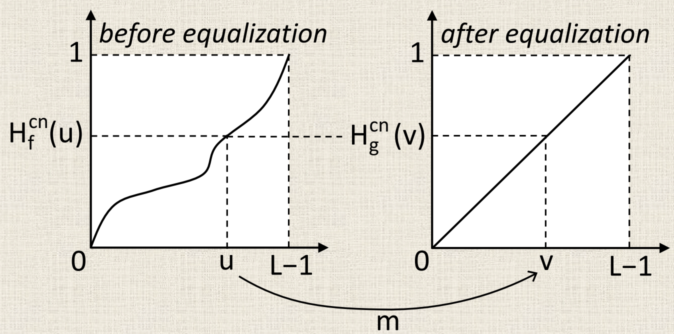
Prof. Pascal Matsakis

II.4c. Histogram Equalization



Prof. Pascal Matsakis

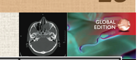
II.4d. Histogram Equalization

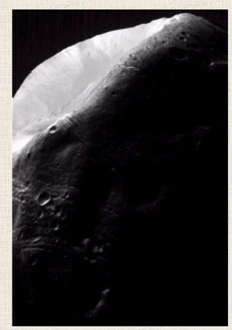


$$\left.\begin{array}{l} H_g^{cn}(v)=H_f^{cn}(u) \\ v=m(u) \\ H_g^{cn}(v)=v/(L-1) \end{array}\right\} \ \mathbf{m(u) = (L\text{-}1)\ H_f^{cn}(u)}$$

Prof. Pascal Matsakis

II.4e. Histogram Equalization

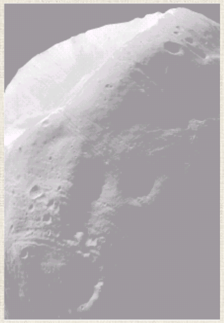


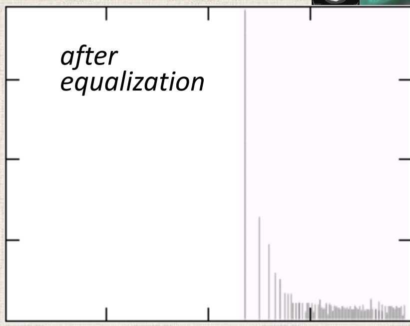


example

Prof. Pascal Matsakis

II.4f. Histogram Equalization





example

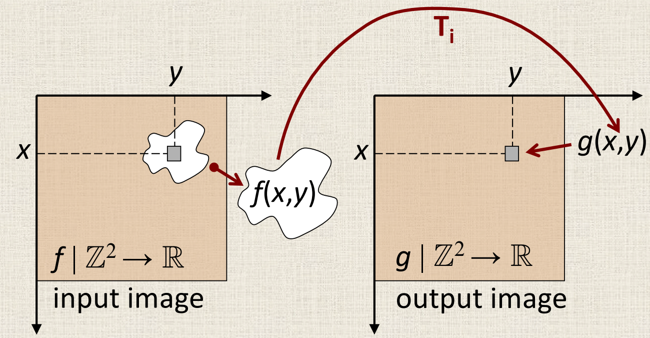
In practice:
- H^{cn} looks linear but H^{n} does not look flat
- the improvement is optimal statistically, not necessarily perceptually.

Prof. Pascal Matsakis

D

Neighborhood Operations

Prof. Pascal Matsakis



$g(x,y)$ depends on the gray levels from some neighborhood of (x,y) in f.

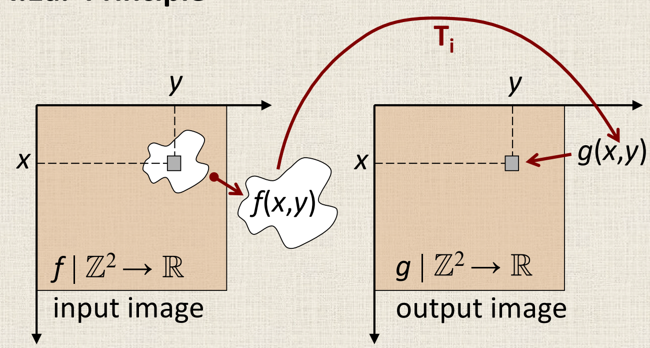
Prof. Pascal Matsakis

Neighborhood Operations

I. Convolution

Prof. Pascal Matsakis

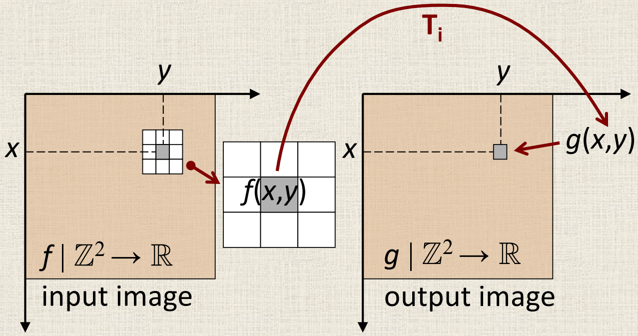
I.1a. Principle



$g(x,y)$ is a **weighted sum** of gray levels from some neighborhood of (x,y) in f.

Prof. Pascal Matsakis

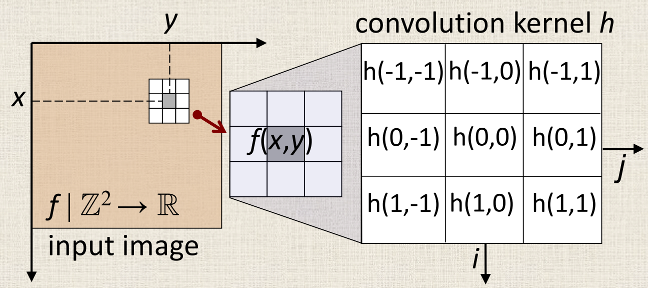
I.1b. Principle



Usually, the neighborhood is an *m*x*n* rectangle (e.g., 1x3, 3x3, 1x5, 5x5) centered at (*x*,*y*).

Prof. Pascal Matsakis

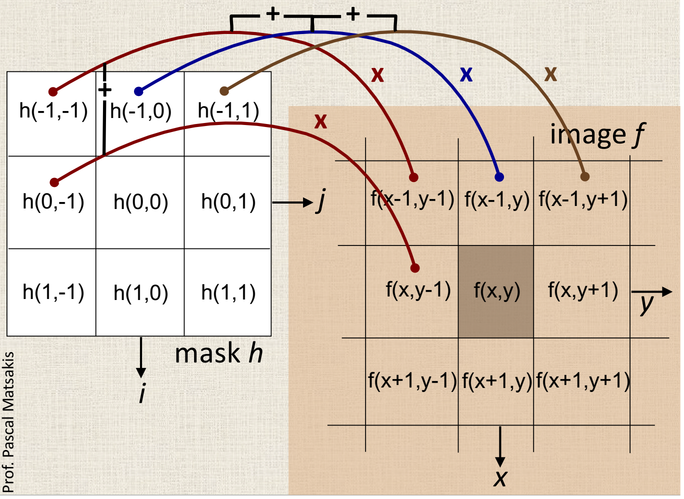
I.1c. Principle



The gray levels are weighted coefficients that come from a **convolution kernel** (or **mask**) *h*.

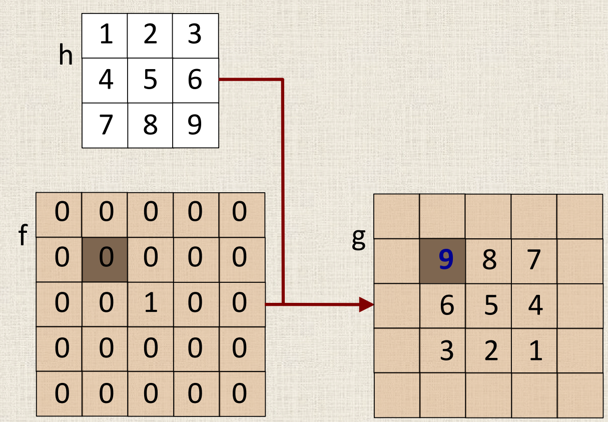
Prof. Pascal Matsakis

I.2a. Definition (1st Attempt)



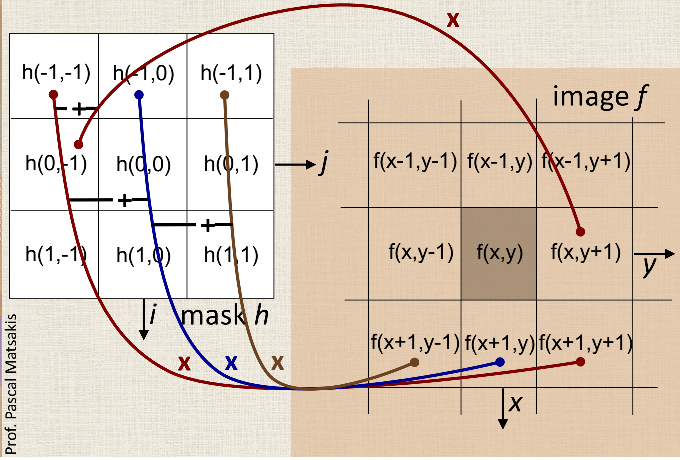
Prof. Pascal Matsakis

I.2b. Definition (1st Attempt)



Prof. Pascal Matsakis

I.3a. Definition (2nd Attempt)



Prof. Pascal Matsakis

I.3b. Definition (2nd Attempt)

h

1	2	3
4	5	6
7	8	9

f

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

g

	1	2	3	
	4	5	6	
	7	8	9	

Prof. Pascal Matsakis

I.3c. Definition (2nd Attempt)

Consider an $m \times n$ kernel h and an image f.
Consider the image g defined by:

$$\underset{h*f}{g(x,y)} = \underbrace{\sum_{i=-\frac{m-1}{2}}^{\frac{m-1}{2}} \sum_{j=-\frac{n-1}{2}}^{\frac{n-1}{2}} h(i,j)\, f(x-i, y-j)}_{(h*f)(x,y)}$$

$h*f$ is the **convolution** of h with f

Prof. Pascal Matsakis

I.4a. Examples

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Prof. Pascal Matsakis

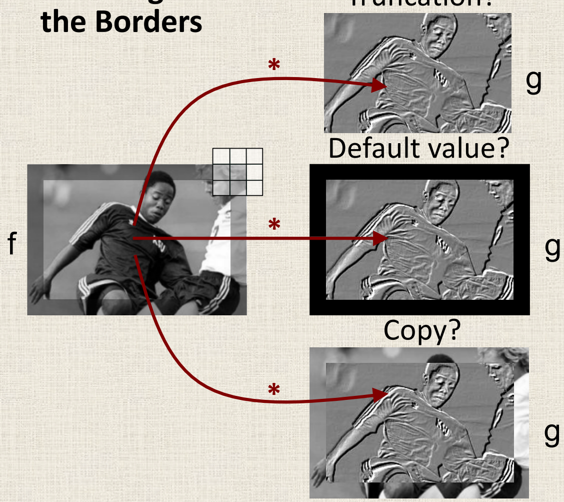
I.4b. Examples

1	1	1
1	1	1
1	1	1


I.4c. Examples

2	0	0
0	-1	0
0	0	-1


I.5a. Handling of the Borders

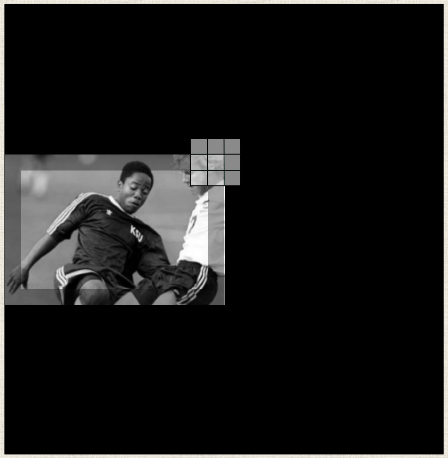


I.5b. Handling of the Borders

Truncation of the kernel?

I.5c. Handling of the Borders

f



Zero padding?

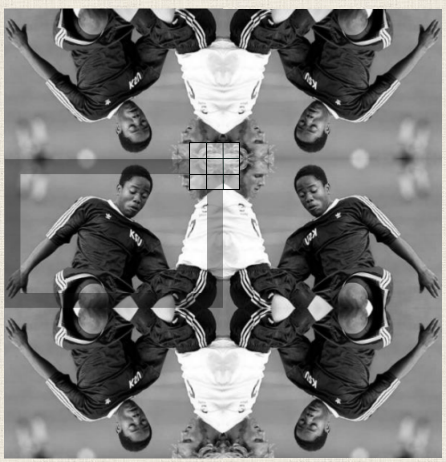
I.5d. Handling of the Borders

f

Circular indexing?

I.5e. Handling of the Borders

f



Reflected indexing?

I.6a. Separable Kernels

Consider an mxn kernel h.

h is **separable** iff
there exist an mx1 kernel h_1 and a 1xn kernel h_2
such that $h = h_1 * h_2$

h is **separable** iff
there exist an mx1 kernel h_1 and a 1xn kernel h_2
such that $h = h_1 h_2$

I.6b. Separable Kernels

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad [1\ 1\ 1] \times [1\ 1\ 1]^T$$

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad [-1\ 0\ 1] \times [-2\ -1\ 0]^T$$

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad \text{no}$$

Are these kernels separable?

I.6c. Separable Kernels

Consider a separable *mxn* kernel $h = h_1 * h_2$

We have: $\mathbf{(h_1 * h_2) * f = h_1 * (h_2 * f)}$

mn multiplications
mn-1 additions

convolution at a single pixel

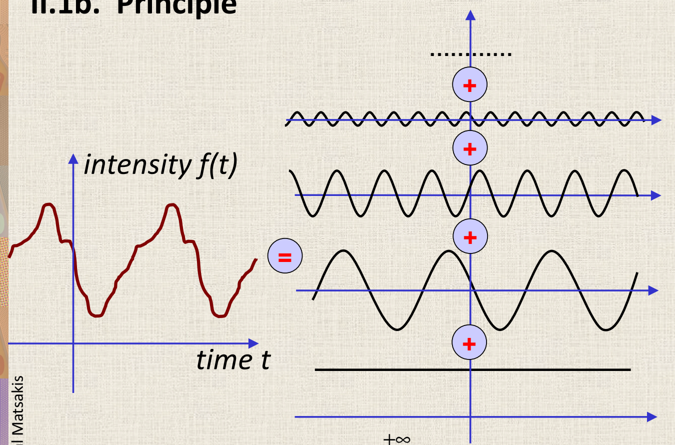
m+*n* multiplications
m+*n*-2 additions

Neighborhood Operations

II. Filtering

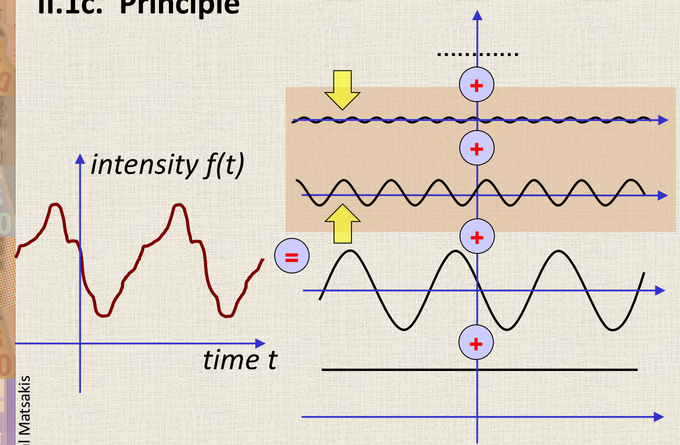
II.1a. Principle

II.1b. Principle



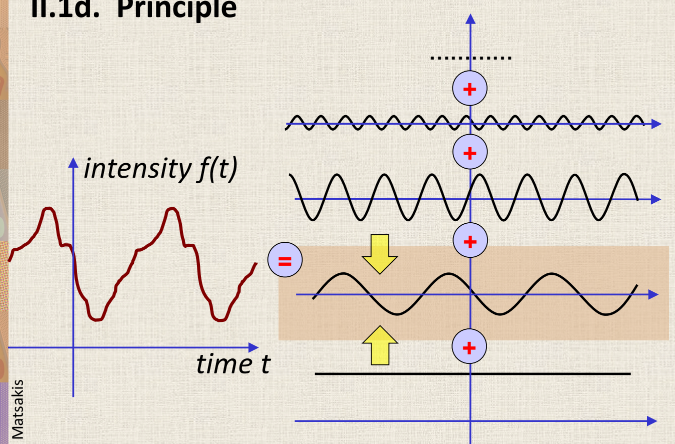
Fourier series: $f(t)=\sum_{u=0}^{+\infty} c_u \cos(\omega u t+\varphi_u)$

II.1c. Principle



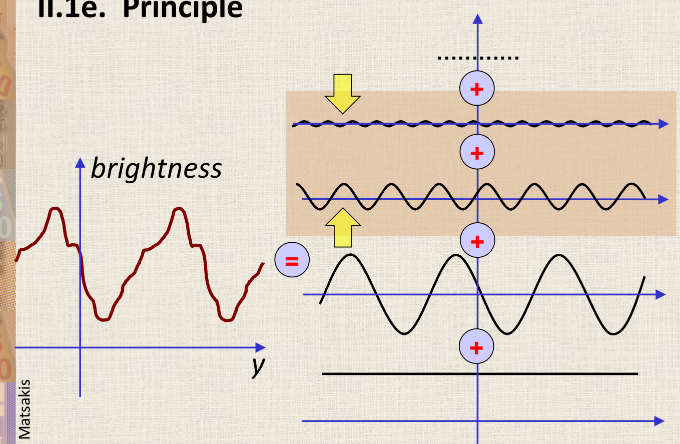
"turn down the treble" = low-pass filter

II.1d. Principle



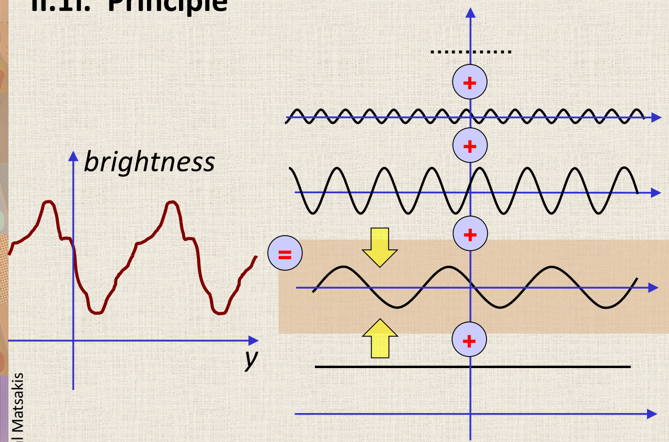
"turn down the bass" = high-pass filter

II.1e. Principle



"blur the image" = low-pass filter

II.1f. Principle



"sharpen the image" = high-pass filter

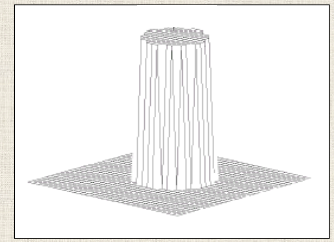
II.2a. Linear Low-Pass Filtering

Convolution kernel with positive coefficients

Example:
mean filtering (uniform)

1	1	1
1	1	1
1	1	1

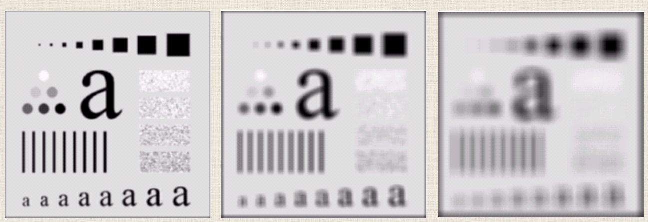
3x3 mean mask
(separable)



II.2b. Linear Low-Pass Filtering

Convolution kernel with positive coefficients

Example:
mean filtering (uniform)



5x5 15x15 35x35

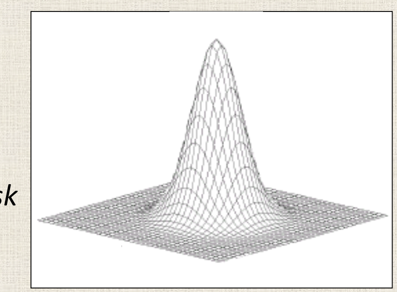
II.2c. Linear Low-Pass Filtering

Convolution kernel with positive coefficients

Example:
Gaussian filtering (non-uniform)

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

5x5 Gaussian mask
(separable)



II.3a. Linear High-Pass Filtering

Convolution kernel with mixture of positive and negative coefficients

Example:
Laplacian filtering (omnidirectional)

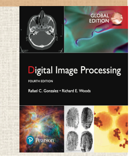
0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

3x3 Laplacian masks

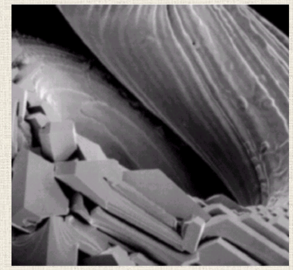
Prof. Pascal Matsakis

II.3b. Linear High-Pass Filtering

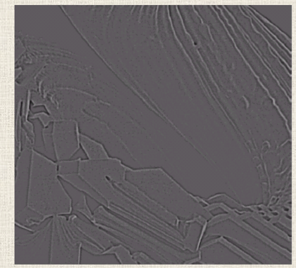


Convolution kernel with mixture of positive and negative coefficients

Example:
Laplacian filtering (omnidirectional)



input



output

Prof. Pascal Matsakis

II.3c. Linear High-Pass Filtering

Example: **unsharp masking**
Subtracting from image output from low pass filter

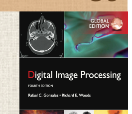
Example: **high-boost filtering**
Adding to image output from high pass filter

0	−1	0
−1	$A + 4$	−1
0	−1	0

−1	−1	−1
−1	$A + 8$	−1
−1	−1	−1

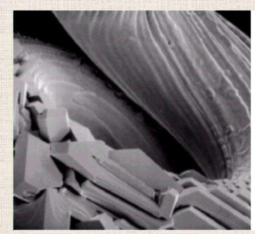
Prof. Pascal Matsakis

II.3d. Linear High-Pass Filtering

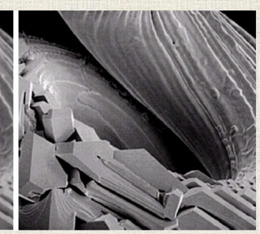


Example: **unsharp masking**
Subtracting from image output from low pass filter

Example: **high-boost filtering**
Adding to image output from high pass filter



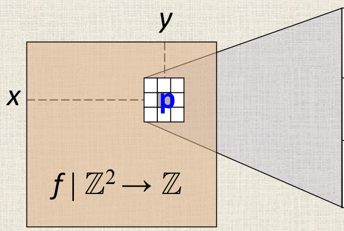
input



output (A=1)

Prof. Pascal Matsakis

II.4a. Order Statistic (Non-Linear) Filtering



1.

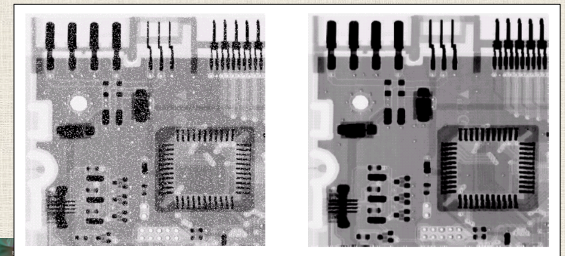
24	221	124
167	**96**	17
80	182	159

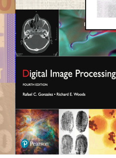
2. (17, 24, 80, 96, 124, 159, 167, 182, 221)

3. g(p) = (value in a specific position)

II.4b. Order Statistic (Non-Linear) Filtering

X-ray image of a circuit board corrupted by salt noise

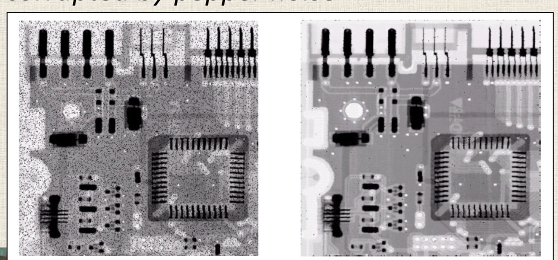


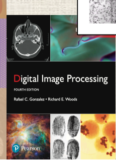


*3x3 **min filtering***

II.4c. Order Statistic (Non-Linear) Filtering

X-ray image of a circuit board corrupted by pepper noise

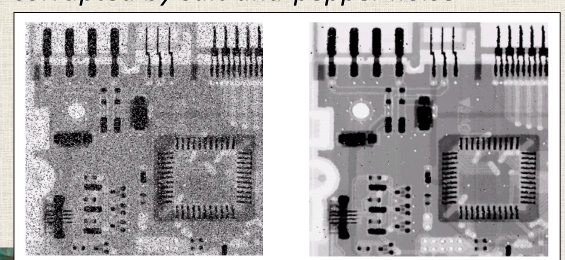


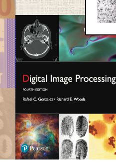


*3x3 **max filtering***

II.4d. Order Statistic (Non-Linear) Filtering

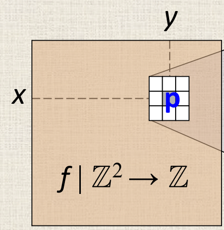
X-ray image of a circuit board corrupted by salt-and-pepper noise





*3x3 **median filtering***

II.5a. α-Trimmed Mean (Non-Linear) Filtering



1.

24	221	124
167	**96**	17
80	182	159

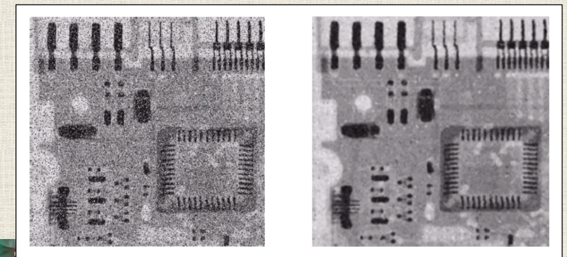
2. (17, 24, 80, 96, 124, 159, 167, 182, 221)

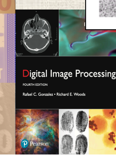
3. (~~17, 24~~, 80, 96, 124, 159, 167, ~~182, 221~~)
α values — g(p) = *average* — *α values*

Prof. Pascal Matsakis

II.5b. α-Trimmed Mean (Non-Linear) Filtering

X-ray image of a circuit board corrupted by uniform noise and salt-and-pepper noise

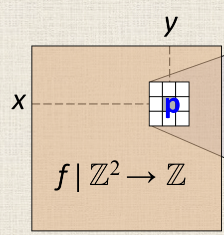




5x5 ***α-trimmed mean filtering***

Prof. Pascal Matsakis

II.6a. Adaptive (Non-Linear) Filtering



1.

24	221	124
167	**96**	17
80	182	159

noise variance: V_n

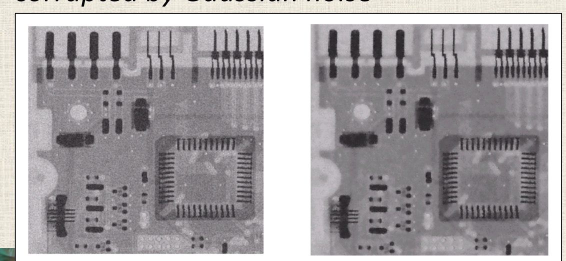
2. mean gray level: m(p)
variance: V(p)

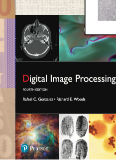
3. $g(p) = f(p) - [f(p) - m(p)]\, V_n / V(p)$

Prof. Pascal Matsakis

II.6b. Adaptive (Non-Linear) Filtering

X-ray image of a circuit board corrupted by Gaussian noise





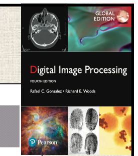
7x7 ***adaptive filtering***

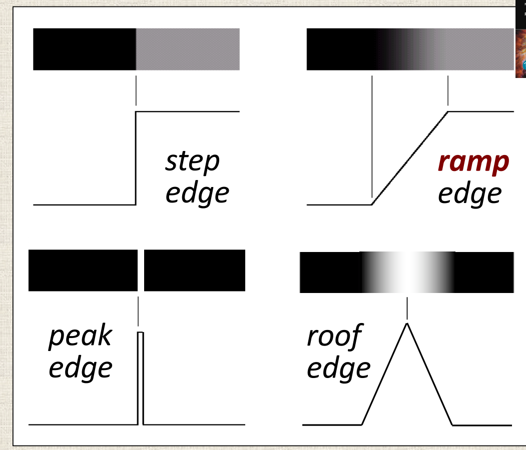
Prof. Pascal Matsakis

E

Edge Detection

Prof. Pascal Matsakis





Prof. Pascal Matsakis

Step 1 = noise reduction
suppress as much noise as possible,
without smoothing away the meaningful edges

Step 2 = edge enhancement
apply some filter that responds strongly at edges,
weakly elsewhere

Step 3 = edge localization
decide which of the values output by the filter
correspond to meaningful edges

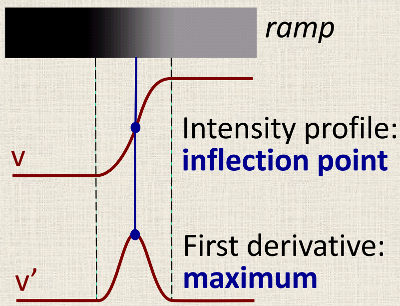
Prof. Pascal Matsakis

Edge Detection

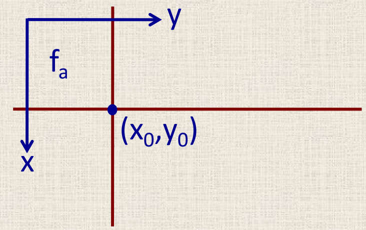
I. Image Gradient

Prof. Pascal Matsakis

I.1. Principle



I.2a. Definitions



$$v = f_a(x_0,\bullet)$$

$$v' = \frac{\mathrm{d}f_a(x_0,\bullet)}{\mathrm{d}y}$$

$$v'(y_0) = \frac{\partial f_a}{\partial y}(x_0,y_0)$$

$$u = f_a(\bullet,y_0)$$

$$u' = \frac{\mathrm{d}f_a(\bullet,y_0)}{\mathrm{d}x}$$

$$u'(x_0) = \frac{\partial f_a}{\partial x}(x_0,y_0)$$

I.2b. Definitions

$$\vec{\nabla} f_a = \left(\frac{\partial f_a}{\partial x}, \frac{\partial f_a}{\partial y}\right)$$

gradient vector

$$G = \sqrt{\cancel{\left(\frac{\partial f_a}{\partial x}\right)^2 + \left(\frac{\partial f_a}{\partial y}\right)^2}}$$

$$G = \left|\frac{\partial f_a}{\partial x}\right| + \left|\frac{\partial f_a}{\partial y}\right|$$

gradient magnitude
edge corresponds to local maximum

$$\theta = \arctan\left(\frac{\partial f_a}{\partial y} \Big/ \frac{\partial f_a}{\partial x}\right)$$

gradient direction
edge runs in perpendicular direction

I.3a. Estimation

$$u'(x_0) = \lim_{h\to 0} \frac{u(x_0+h)-u(x_0)}{h}$$

$$u'(x_0) = \lim_{h\to 0} \frac{u(x_0+h)-u(x_0-h)}{2h}$$

For $h \approx 0$: $\quad u'(x_0) \approx \dfrac{u(x_0+h)-u(x_0-h)}{2h}$

$$\frac{\partial f_a}{\partial x}(x_0,y_0) \approx \frac{f_a(x_0+h,y_0)-f_a(x_0-h,y_0)}{2h}$$

Choose $h=1$, assume $x_0 \in \mathbb{Z}$ and $y_0 \in \mathbb{Z}$:

$$\frac{\partial f_a}{\partial x}(x_0,y_0) \approx \frac{f(x_0+1,y_0)-f(x_0-1,y_0)}{2}$$

I.3b. Estimation

$$\frac{\partial f_a}{\partial x}(x_0,y_0) \approx \frac{f(x_0+1,y_0)-f(x_0-1,y_0)}{2}$$

$$\frac{\partial f_a}{\partial x} \approx \begin{bmatrix}1/2\\0\\-1/2\end{bmatrix} * f$$

$$\frac{\partial f_a}{\partial y}(x_0,y_0) \approx \frac{f(x_0,y_0+1)-f(x_0,y_0-1)}{2}$$

$$\frac{\partial f_a}{\partial y} \approx \begin{bmatrix}1/2 & 0 & -1/2\end{bmatrix} * f$$

I.4a. Prewitt Kernels

Noise reduction:

$[\,1\ 1\ 1\,] * f$ .... $\begin{bmatrix}1\\1\\1\end{bmatrix} * f$

Edge enhancement:

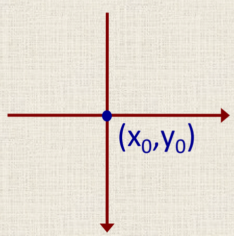
$\begin{bmatrix}-1\\0\\1\end{bmatrix} * ([1\ 1\ 1] * f)$ .... $[-1\ \ 0\ \ 1] * (\begin{bmatrix}1\\1\\1\end{bmatrix} * f)$

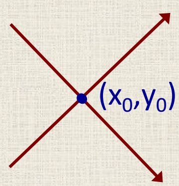
Edge localization:

$\frac{\partial f_a}{\partial x} \simeq \begin{bmatrix}-1 & -1 & -1\\0 & 0 & 0\\1 & 1 & 1\end{bmatrix} * f$ .... $\frac{\partial f_a}{\partial y} \simeq \begin{bmatrix}-1 & 0 & 1\\-1 & 0 & 1\\-1 & 0 & 1\end{bmatrix} * f$

Binary **edge map**: gray level of *p* is 255 iff *G*(*p*)>τ
(where *G*(*p*) is the gradient magnitude at *p*)

I.4b. Prewitt Kernels





−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

−1	−1	0
−1	0	1
0	1	1

0	1	1
−1	0	1
−1	−1	0

I.5a. Sobel Kernels

Noise reduction:

$[\,1\ 2\ 1\,] * f$ .... $\begin{bmatrix}1\\2\\1\end{bmatrix} * f$

Edge enhancement:

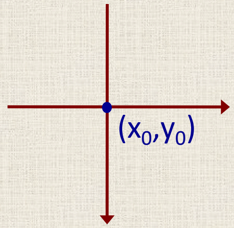
$\begin{bmatrix}-1\\0\\1\end{bmatrix} * ([1\ 2\ 1] * f)$ .... $[-1\ \ 0\ \ 1] * (\begin{bmatrix}1\\2\\1\end{bmatrix} * f)$

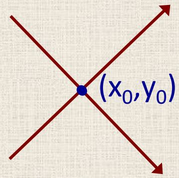
Edge localization:

$\frac{\partial f_a}{\partial x} \simeq \begin{bmatrix}-1 & -2 & -1\\0 & 0 & 0\\1 & 2 & 1\end{bmatrix} * f$ .... $\frac{\partial f_a}{\partial y} \simeq \begin{bmatrix}-1 & 0 & 1\\-2 & 0 & 2\\-1 & 0 & 1\end{bmatrix} * f$

Binary **edge map**: gray level of *p* is 255 iff *g*(*p*)>τ
(where *g*(*p*) is the gradient magnitude at *p*)

I.5b. Sobel Kernels





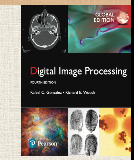
−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

−2	−1	0
−1	0	1
0	1	2

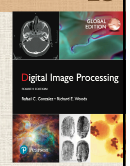
0	1	2
−1	0	1
−2	−1	0

I.5c. Sobel Kernels

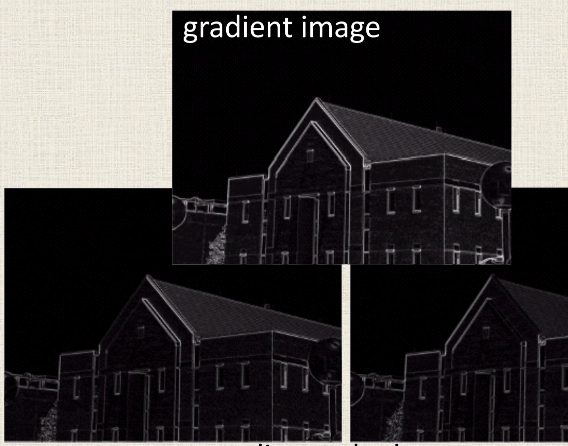


horizontal and vertical edges

I.5d. Sobel Kernels

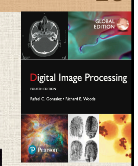


gradient image

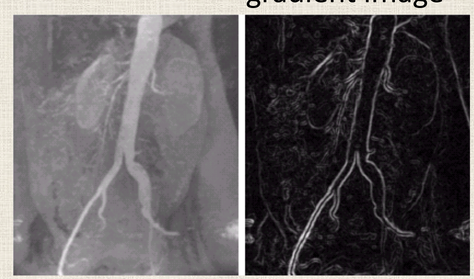


diagonal edges

I.5e. Sobel Kernels



gradient image



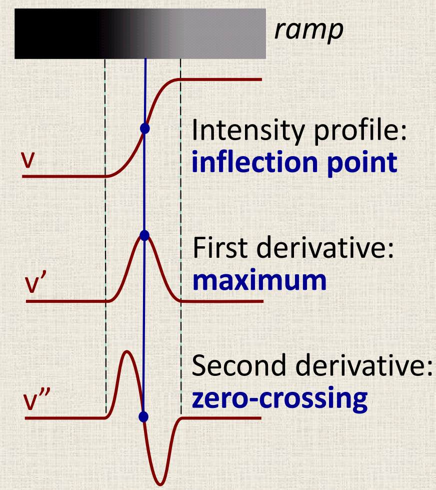
aortic angiogram (X-ray image)

Edge Detection

II. Image Laplacian

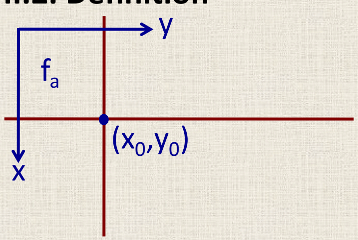
Prof. Pascal Matsakis

II.1. Principle



Prof. Pascal Matsakis

II.2. Definition



$$v = f_a(x_0,\bullet)$$

$$v'' = \frac{\mathrm{d}^2 f_a(x_0,\bullet)}{\mathrm{d}y^2}$$

$$v''(y_0) = \frac{\partial^2 f_a}{\partial y^2}(x_0,y_0)$$

$$u = f_a(\bullet,y_0)$$

$$u'' = \frac{\mathrm{d}^2 f_a(\bullet,y_0)}{\mathrm{d}x^2}$$

$$u''(x_0) = \frac{\partial^2 f_a}{\partial x^2}(x_0,y_0)$$

$$\underbrace{\nabla^2 f_a = \frac{\partial^2 f_a}{\partial x^2} + \frac{\partial^2 f_a}{\partial y^2}}_{\text{Laplacian}}$$

Prof. Pascal Matsakis

II.3a. Estimation

For $h \approx 0$: $u''(x_0) \approx \dfrac{u'(x_0+h) - u'(x_0-h)}{2h}$

For $k \approx 0$: $u'(x_0+h) \approx \dfrac{u((x_0+h)+k) - u((x_0+h)-k)}{2k}$

For $l \approx 0$: $u'(x_0-h) \approx \dfrac{u((x_0-h)+l) - u((x_0-h)-l)}{2l}$

Choose $h = k = l$:

$$\frac{\partial^2 f_a}{\partial x^2}(x_0,y_0) \approx \frac{f_a(x_0+2h,y_0) - 2f_a(x_0,y_0) + f_a(x_0-2h,y_0)}{4h^2}$$

Choose $h = 1/2$, assume $x_0 \in \mathbb{Z}$ and $y_0 \in \mathbb{Z}$:

$$\frac{\partial^2 f_a}{\partial x^2}(x_0,y_0) \approx f(x_0+1,y_0) - 2f(x_0,y_0) + f(x_0-1,y_0)$$

Prof. Pascal Matsakis

II.3b. Estimation

$$\frac{\partial^2 f_a}{\partial x^2}(x_0,y_0) \approx f(x_0+1,y_0) - 2f(x_0,y_0) + f(x_0-1,y_0)$$

$$\frac{\partial^2 f_a}{\partial x^2} \approx \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} * f$$

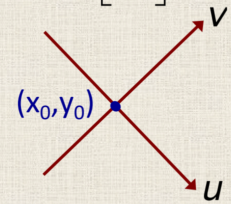
$$\frac{\partial^2 f_a}{\partial y^2}(x_0,y_0) \approx f(x_0,y_0+1) - 2f(x_0,y_0) + f(x_0,y_0-1)$$

$$\frac{\partial^2 f_a}{\partial y^2} \approx \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} * f$$

Prof. Pascal Matsakis

II.4a. Laplacian Kernels

$$\nabla^2 f \approx \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} * f + \begin{bmatrix} 1 & -2 & 1 \end{bmatrix} * f = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} * f$$



$$\nabla^2 f \approx \begin{bmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{bmatrix} * f$$

$$\nabla^2 f \approx \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} * f$$

Prof. Pascal Matsakis

II.4b. Laplacian Kernels

Noise reduction:

none

Edge enhancement:

$K_L * f$ where K_L is a Laplacian kernel

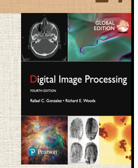
Edge localization:

FOR each pixel *p* **DO**
    FOR each 8-neighbour *q* of *p* **DO**
        IF $L(q)L(q')<0$ **and** $|L(q)-L(q')|>\tau$
        % where L(q) is the Laplacian at q and
        % q' is the neighbour of p opposite to q
        THEN mark *p* as an edge
        % set its gray level to 255 in the ***edge map***

threshold so we take meaningful edges

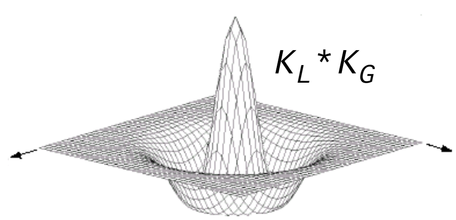
Prof. Pascal Matsakis

II.5a. Laplacian of Gaussian



Noise reduction: $K_G * f$
where K_G is a Gaussian kernel

Edge enhancement: $K_L * (K_G * f) = (K_L * K_G) * f$
where K_L is a Laplacian kernel



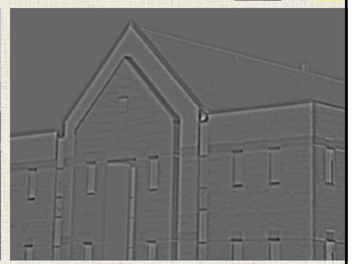
0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

Edge localization: replace *L* with *LoG*

Prof. Pascal Matsakis

II.5b. Laplacian of Gaussian

input image f



LoG $K_L * (K_G * f)$

Prof. Pascal Matsakis

II.5c. Laplacian of Gaussian

binary edge map
(threshold τ=0%)

binary edge map
(threshold τ=4%)

Prof. Pascal Matsakis

II.5d. Laplacian of Gaussian

aortic angiogram
(X-ray image)

LoG

Prof. Pascal Matsakis

II.5e. Laplacian of Gaussian

thresholded LoG

this is just LoG not thresholded

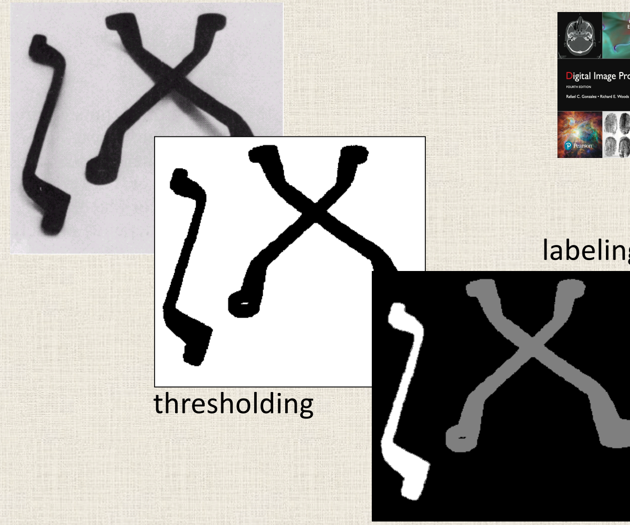
binary edge map

Prof. Pascal Matsakis

F

Thresholding and Labeling

Prof. Pascal Matsakis



thresholding

labeling

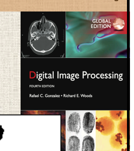
Prof. Pascal Matsakis

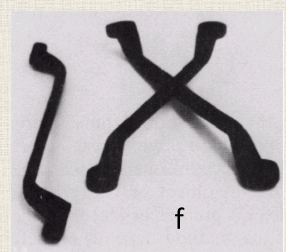
Thresholding and Labeling

I. Thresholding

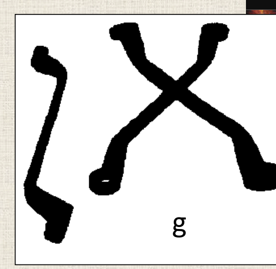
Prof. Pascal Matsakis

I.1. Principle





f

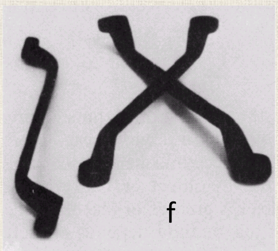


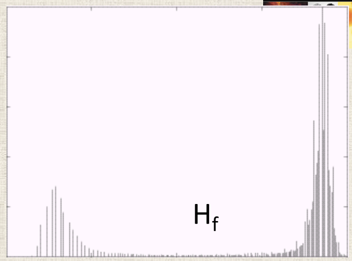
g

If $f(x,y) \geq \tau$ then $g(x,y)=255$
else $g(x,y)=0$

Prof. Pascal Matsakis

I.2a. Choosing the Threshold





τ derived from H_f

Prof. Pascal Matsakis

I.2b. Choosing the Threshold

basic global thresholding

$u_{new} \leftarrow$ average f(x,y)
do
- $u_{old} \leftarrow u_{new}$
- $u_{inf} \leftarrow$ average f(x,y) with $f(x,y) \le u_{old}$
- $u_{max} \leftarrow$ average f(x,y) with $f(x,y) > u_{old}$
- $u_{new} \leftarrow 0.5(u_{inf}+u_{max})$

until $|u_{new}-u_{old}| \le \delta$

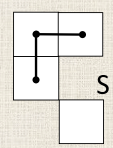
Prof. Pascal Matsakis

Thresholding and Labeling

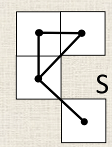
II. Connected Components

Prof. Pascal Matsakis

II.1. Adjacency



4-adjacency in S

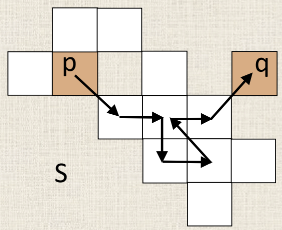


8-adjacency in S

p is **adjacent** to q in S iff?
S_1 is **adjacent** to S_2 iff?

Prof. Pascal Matsakis

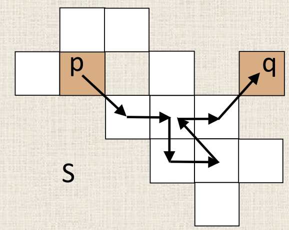
II.2. Paths



8-path in S, of length 7, from p to q

Π is a **path** in S iff?

II.3a. Connectivity



p is 8-connected to q in S
p is not 4-connected to q in S

p is **connected** to q in S iff?
S is **connected** iff?

II.3b. Connectivity

Binary relation on S defined by
"*p is related to q iff p is connected to q in S*"
is an **equivalence relation**.

The **equivalence classes** are
the **connected components** of S.

They define a **partition** of S.

Thresholding and Labeling

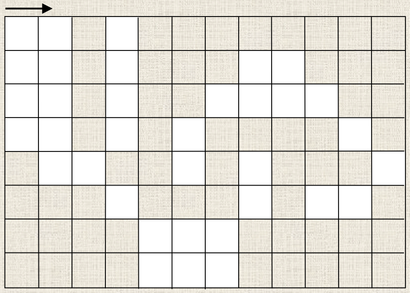
III. Labeling

III.1. Build a Binary Relation

How to find the set of 8-connected components?

STEP 1

scan →



Relation $\Re$

$$\begin{bmatrix} \; \end{bmatrix}$$

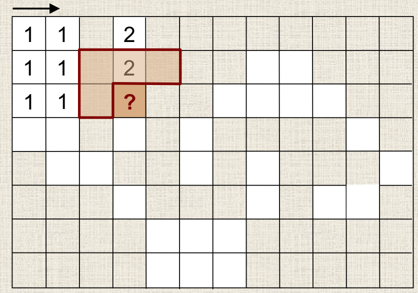
Prof. Pascal Matsakis

III.1. Build a Binary Relation

How to find the set of 8-connected components?

STEP 1

scan →



Relation $\Re$

$$\begin{array}{c|cc} & 1 & 2 \\ \hline 1 & 1 & 0 \\ 2 & 0 & 1 \end{array}$$

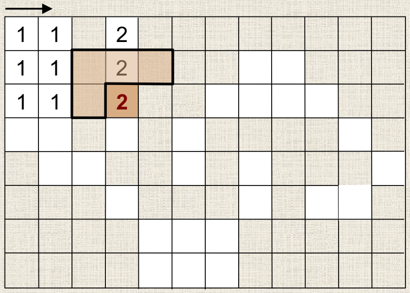
Prof. Pascal Matsakis

III.1. Build a Binary Relation

How to find the set of 8-connected components?

STEP 1

scan →



Relation $\Re$

$$\begin{array}{c|cc} & 1 & 2 \\ \hline 1 & 1 & 0 \\ 2 & 0 & 1 \end{array}$$

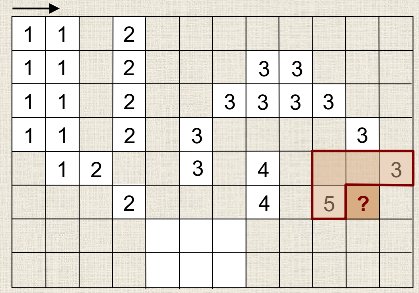
Prof. Pascal Matsakis

III.1. Build a Binary Relation

How to find the set of 8-connected components?

STEP 1

scan →



Relation $\Re$

$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 & 0 \\ 4 & 0 & 0 & 0 & 1 & 0 \\ 5 & 0 & 0 & 0 & 0 & 1 \end{array}$$

Prof. Pascal Matsakis

III.1. Build a Binary Relation

How to find the set of 8-connected components?

STEP 1

scan →

1	1		2								
1	1		2				3	3			
1	1		2			3	3	3	3		
1	1		2		3					3	
	1	2			3		4				3
			2				4		5	3	

Relation ℜ

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 & 0 \\ 3 & 0 & 0 & 1 & 0 & 1 \\ 4 & 0 & 0 & 0 & 1 & 0 \\ 5 & 0 & 0 & 1 & 0 & 1 \end{array}$$

Prof. Pascal Matsakis

III.1. Build a Binary Relation

How to find the set of 8-connected components?

STEP 1

1	1		2								
1	1		2				3	3			
1	1		2			3	3	3	3		
1	1		2		3					3	
	1	2			3		4				3
			2				4		5	3	
				2	2	4					
				2	4	4					

→ *scan*

Relation ℜ

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 & 0 & 1 \\ 4 & 0 & 1 & 0 & 1 & 0 \\ 5 & 0 & 0 & 1 & 0 & 1 \end{array}$$

Prof. Pascal Matsakis

III.2. Calculate Its Transitive Closure

How to find the set of 8-connected components?

STEP 2 (Warshall Algorithm)

Relation ℜ

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 & 0 & 1 \\ 4 & 0 & 1 & 0 & 1 & 0 \\ 5 & 0 & 0 & 1 & 0 & 1 \end{array}$$

Prof. Pascal Matsakis

III.2. Calculate Its Transitive Closure

How to find the set of 8-connected components?

STEP 2 (Warshall Algorithm)

Transitive closure
$\Re^+$ of ℜ

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 & 0 & 1 \\ 4 & 1 & 1 & 0 & 1 & 0 \\ 5 & 0 & 0 & 1 & 0 & 1 \end{array}$$

Prof. Pascal Matsakis

III.2. Calculate Its Transitive Closure

How to find the set of 8-connected components?

STEP 2 (Warshall Algorithm)

$\Re^+ = \Re$

% $\Re^+$ is now [r_{ij}]

for k=1 **to** n

  for i=1 **to** n

    for j=1 **to** n

      $r_{ij}=\max\{r_{ij},\min\{r_{ik},r_{kj}\}\}$

      % if r_{ik} and r_{kj} then $r_{ij}=1$

      % else keep r_{ij} as is

III.3. Calculate the Equivalence Classes

How to find the set of 8-connected components?

STEP 3

Transitive closure $\Re^+$ of $\Re$

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 1 & 1 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 & 0 & 1 \\ 4 & 1 & 1 & 0 & 1 & 0 \\ 5 & 0 & 0 & 1 & 0 & 1 \end{array}$$

III.3. Calculate the Equivalence Classes

How to find the set of 8-connected components?

STEP 3

Transitive closure $\Re^+$ of $\Re$

equivalence classes

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 1 & 1 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 & 0 & 1 \\ 4 & 1 & 1 & 0 & 1 & 0 \\ 5 & 0 & 0 & 1 & 0 & 1 \end{array}$$

III.4. Label the Connected Components

How to find the set of 8-connected components?

STEP 4

scan →

1	1		2								
1	1		2				3	3			
1	1		2			3	3	3	3		
1	1		2		3					3	
	1	2			3		4				3
			2				4		5	3	
				2	2	4					
				2	4	4					

Equivalence classes of $\Re^+$

$$\begin{array}{c|ccccc|} & 1 & 2 & 3 & 4 & 5 \\ \hline ✓1 & 1 & 1 & 0 & 1 & 0 \\ 2 & 1 & 1 & 0 & 1 & 0 \\ ✓3 & 0 & 0 & 1 & 0 & 1 \\ 4 & 1 & 1 & 0 & 1 & 0 \\ 5 & 0 & 0 & 1 & 0 & 1 \end{array}$$

III.4. Label the Connected Components

How to find the set of 8-connected components?

STEP 4

1	1		**1**								
1	1		**1**				3	3			
1	1		**1**			3	3	3	3		
1	1		**1**		3					3	
	1	**1**			3		**1**				3
			1				**1**		**3**	3	
				1	**1**	**1**					
				1	**1**	**1**					

→ *scan*

Equivalence classes of $\Re^+$

	1	2	3	4	5
✓1	1	1	0	1	0
2	1	1	0	1	0
✓3	0	0	1	0	1
4	1	1	0	1	0
5	0	0	1	0	1

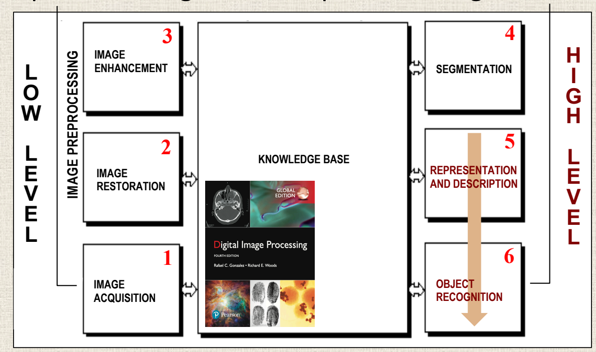
Prof. Pascal Matsakis

Prof. Pascal Matsakis

Towards Object Recognition

Prof. Pascal Matsakis

outputs are images | outputs are image attributes



autonomous machine perception

Prof. Pascal Matsakis

Towards Object Recognition

I. Representation of Image Regions

Prof. Pascal Matsakis

I.1a. Boundary Following

0	0	0	0	0	0	0
0	0	1	1	1	1	0
0	1			1	0	0
0	0	1		1	0	0
0	1			1	0	0
0	1	1	1	1	0	0
0	0	0	0	0	0	0

Goal:
ordering the points in the boundary of a region

I.1b. Boundary Following

	n_0	b_0	1	1	1	
	1			1		
		1		1		
	1			1		
	1	1	1	1		

Initialization:

b_0 = uppermost leftmost point
n_0 = its west neighbour

Prof. Pascal Matsakis

I.1c. Boundary Following

			n_1			
	n_0	b_0	b_1	1	1	
	1			1		
		1		1		
	1			1		
	1	1	1	1		

Starting at n_i and in a clockwise direction:

b_{i+1} = first neighbour of b_i that belongs to boundary
n_{i+1} = neighbour encountered just before b_{i+1}

Prof. Pascal Matsakis

I.1d. Boundary Following

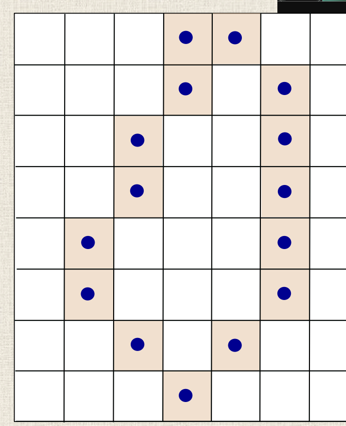
		b_0	b_1	b_2	b_3	
	b_{13}			b_4		
		b_{12}		b_5		
	b_{11}			b_6		
	b_{10}	b_9	b_8	b_7		

Termination: $b_k = b_0$ and $b_{k+1} = b_1$

Result: $(b_0, b_1, b_2, \dots, b_{k-1})$

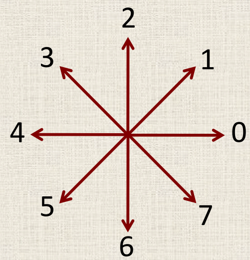
Prof. Pascal Matsakis

I.2a. Chain Codes

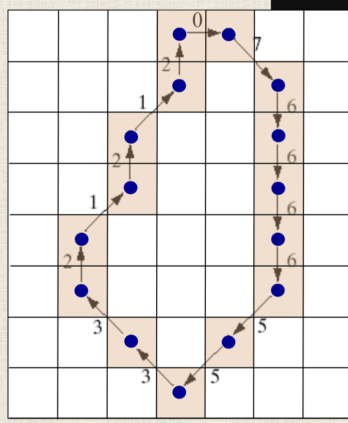


Prof. Pascal Matsakis

I.2b. Chain Codes



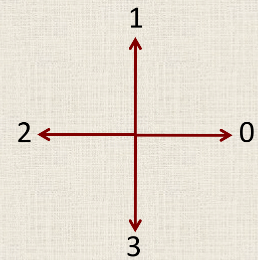
8-directional chain code



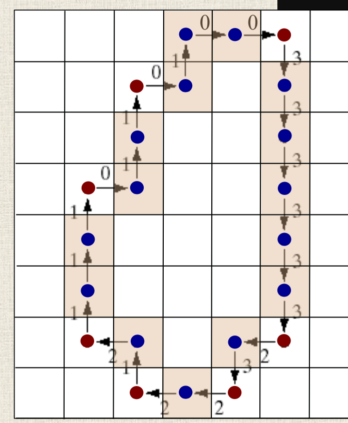
076666553321212

Prof. Pascal Matsakis

I.2c. Chain Codes



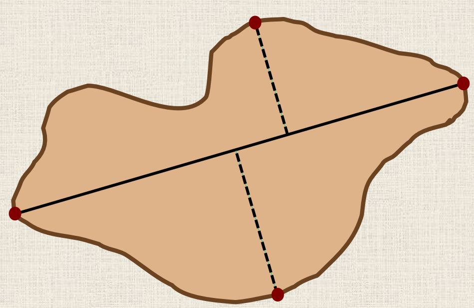
4-directional chain code



003333323221211101101

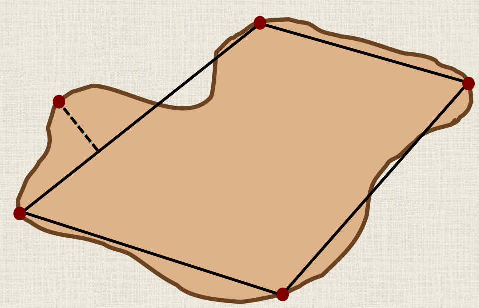
Prof. Pascal Matsakis

I.3a. Polygonal Approximations



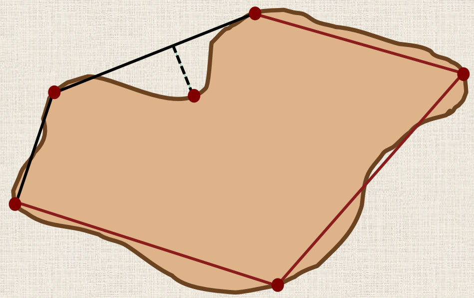
Prof. Pascal Matsakis

I.3b. Polygonal Approximations



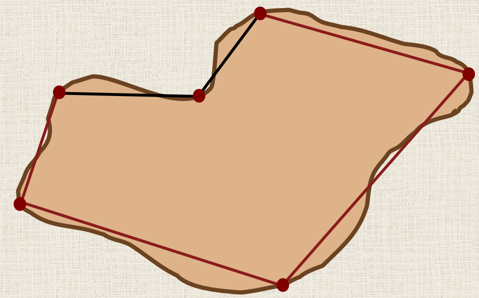
Prof. Pascal Matsakis

I.3c. Polygonal Approximations



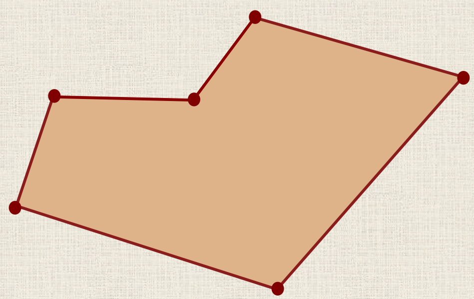
Prof. Pascal Matsakis

I.3d. Polygonal Approximations



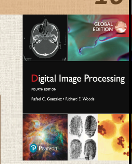
Prof. Pascal Matsakis

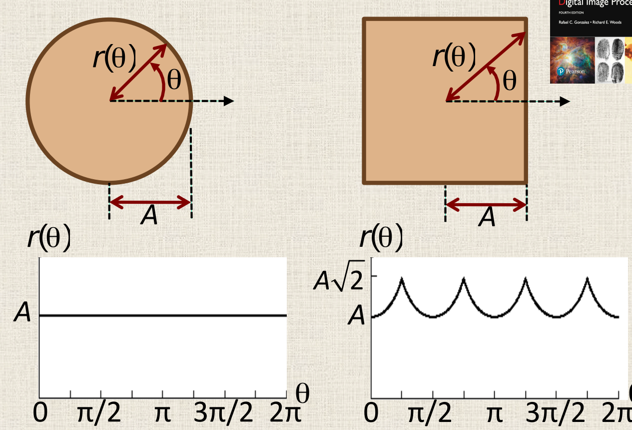
I.3e. Polygonal Approximations



Prof. Pascal Matsakis

I.4. Signatures





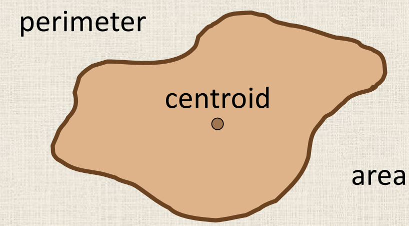
Prof. Pascal Matsakis

Towards Object Recognition

II. Boundary and Regional Descriptors

Prof. Pascal Matsakis

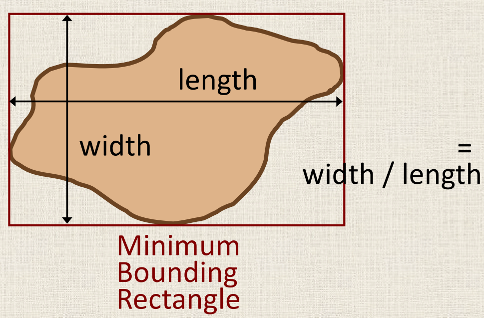
II.1a. Geometrical Descriptors



circularity ratio
$= 4\pi \cdot \text{area} \,/\, \text{perimeter}^2$

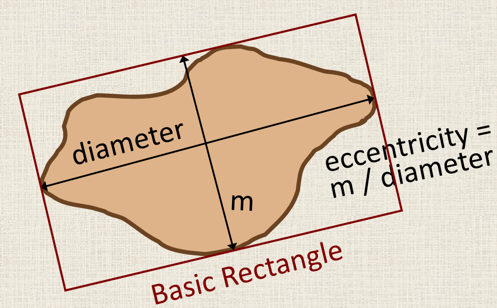
Prof. Pascal Matsakis

II.1b. Geometrical Descriptors



Prof. Pascal Matsakis

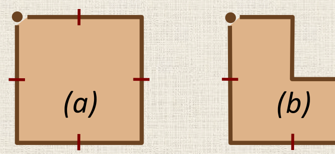
II.1c. Geometrical Descriptors



Prof. Pascal Matsakis

II.2a. Shape Numbers

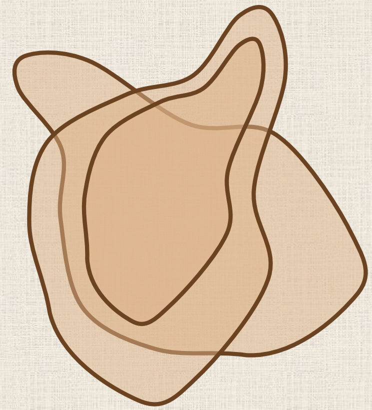
3 shapes of order 8



	(a)	(b)	(c)
chain code	00332211	03032211	00032221
difference	30303030	33133030	30033003
shape number ...	03030303	03033133	00330033

Prof. Pascal Matsakis

II.2b. Shape Numbers

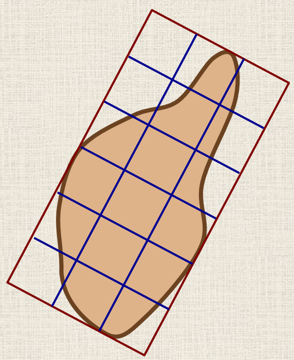


shape number of order 18

?

Prof. Pascal Matsakis

II.2c. Shape Numbers



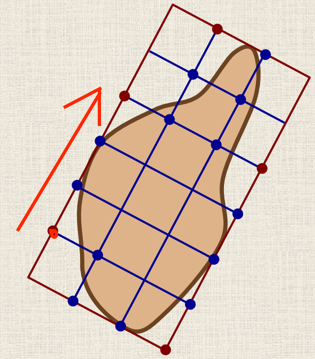
shape number of order 18

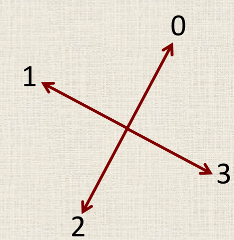
?

3x6 rectangle is rectangle of order **18** whose eccentricity best matches eccentricity of basic rectangle } **3x6** grid!

Prof. Pascal Matsakis

II.2d. Shape Numbers

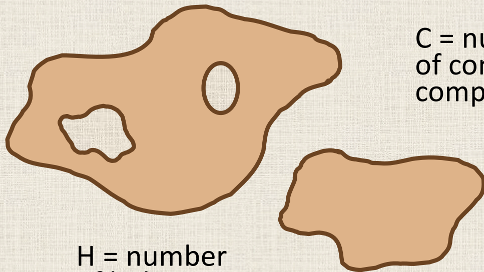




chain code 000300322322221101
difference 300310330130003031
shape number 000303130031033013

Prof. Pascal Matsakis

II.3. Topological Descriptors



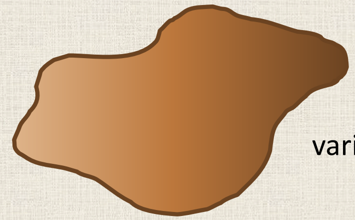
C = number of connected components

H = number of holes

E = C-H = Euler number

Prof. Pascal Matsakis

II.4. Statistical Descriptors

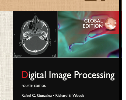


variance of gray level

mean gray level

Prof. Pascal Matsakis

II.5a. Texture



1	1	7	5	3	2
5	1	6	1	2	5
8	8	6	8	1	2
4	3	4	5	5	1
8	7	8	7	6	2
7	8	6	2	6	2

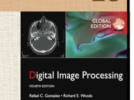
Image f

	1	2	3	4	5	6	7	8
1	1	2	0	0	0	1	1	0
2	0	0	0	0	1	1	0	0
3	0	1	0	1	0	0	0	0
4	0	0	1	0	1	0	0	0
5	2	0	1	0	1	0	0	0
6	1	3	0	0	0	0	0	1
7	0	0	0	0	1	1	0	2
8	1	0	0	0	0	2	2	1

Co-occurrence matrix G
(relation "east neighbour")

Prof. Pascal Matsakis

II.5b. Texture



$$p_{ij} = G_{ij} / \sum G_{kl}$$

Descriptors:

max p_{ij}

energy $\sum p_{ij}^2$

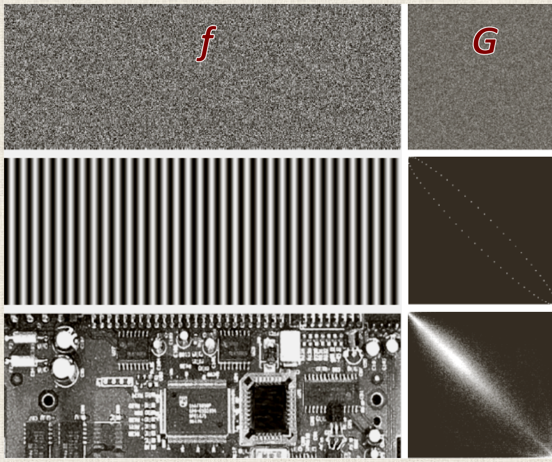
entropy $-\sum p_{ij} \log_2 p_{ij}$

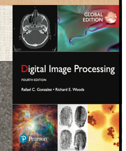
	1	2	3	4	5	6	7	8
1	1	2	0	0	0	1	1	0
2	0	0	0	0	1	1	0	0
3	0	1	0	1	0	0	0	0
4	0	0	1	0	1	0	0	0
5	2	0	1	0	1	0	0	0
6	1	3	0	0	0	0	0	1
7	0	0	0	0	1	1	0	2
8	1	0	0	0	0	2	2	1

Co-occurrence matrix G
(relation "east neighbour")

Prof. Pascal Matsakis

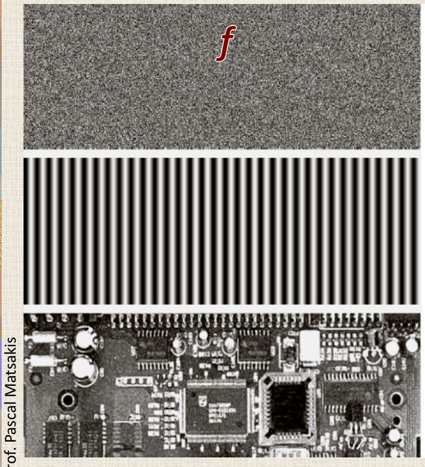
II.5c. Texture





Prof. Pascal Matsakis

II.5d. Texture

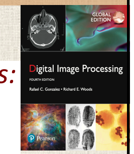


Descriptors:

0.00006
0.00002
15.75

0.01500
0.01230
6.43

0.06860
0.00480
13.58



Prof. Pascal Matsakis

Towards Object Recognition

III. Classification and Clustering

Prof. Pascal Matsakis

III.1a. Terminology

Bezdek
"**Pattern recognition** is a search for structure in data."

Fukunaga
"**Pattern recognition** consists of two parts: feature selection and classifier design."

Devijer and Kittler
"**Pattern recognition** is a very broad field of activities with very fuzzy borders."

Prof. Pascal Matsakis

III.1b. Terminology

patterns
objects

features
descriptors
attributes
characteristics

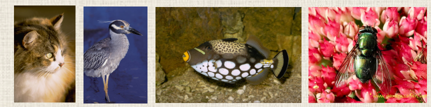
gray level
area
texture
weight
density
wavelength
petal width

classes
clusters

crop
silky
anger
B star
target
male voice
iris Virginica

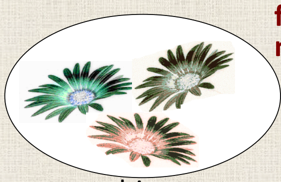
III.1c. Terminology

classification
examine the features of an object
and assign this object to one
of a predefined set of classes



clustering
compare the features of various objects
and assign these objects to different clusters

III.2a. Principle



objects

feature nomination

number of petals
petal width
petal length
petal type
petal texture
sepal width
sepal length

feature selection

number of petals
petal width
petal length
petal texture

III.2b. Principle

given object

feature measurement

$$\begin{pmatrix} 22 \\ 1.2 \\ 4.3 \\ \text{delicate} \end{pmatrix}$$

feature vector
(belongs to the *feature space*)

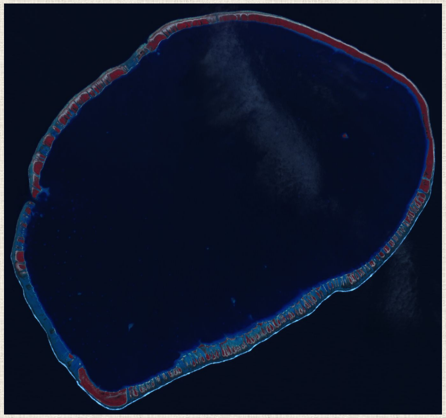
classification or clustering

class or cluster

$$\begin{pmatrix} 0.6 \\ 0.1 \\ 0.3 \end{pmatrix} \quad \begin{matrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \end{matrix}$$

label vector
(can be *crisp*, *probabilistic*, *possibilistic*…)

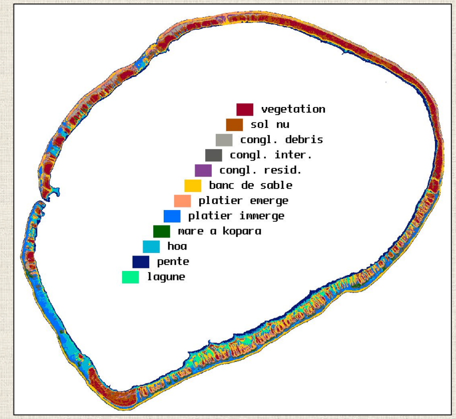
III.3a. Example



input image

Prof. Pascal Matsakis

III.3b. Example

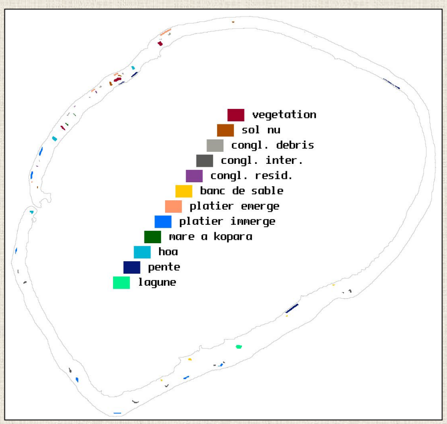


output image

?

Prof. Pascal Matsakis

III.3c. Example



training data

Prof. Pascal Matsakis

III.3d. Example

C classes: 1, 2, ..., k, ..., C

$$m_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_{k_i}$$

where $x_{k_1}, x_{k_2}, \ldots, x_{k_{N_k}}$ are the feature vectors of N_k samples from class k.

Pixel whose feature vector is x is assigned to the class that minimizes $\|x - m_k\|$, i.e., that maximizes the **decision function**

transpose

$$x^t m_k - 0.5 m_k^t m_k.$$

Prof. Pascal Matsakis