1

Enabling Technologies for Sports (5XSF0), Module 01

Motivation, Image Fundamentals and Signal Transformations

Peter H.N. de With

(p.h.n.de.with@tue.nl)

slides version 1.0

2

Set up of this course

* Lectures and instructions
  – 2 x 4 hours / week: 4 hours of teaching + 4 hours instruction
  – Experimenting with algorithms and techniques is essential!
* Study Material
  – Slides lectures & instructions /
  – Prof.dr.ir. Peter H.N. de With / Dr. Sveta Zinger
  – Background book: Digital Image Processing / Gonzales & Woods, Pearson Prentice Hall, 3rd Edition, 2008
  – Contact by email: p.h.n.de.with@tue.nl / s.zinger@tue.nl
  – Website: http://vca.ele.tue.nl/courses/index.html
* **Examn:** Written, and oral afterwards for failures

3

Overview Module 1

* Motivation for this course
  – Applications of sensing and analysis
  – Power of signal and visual sensing
* Signal and 2D Image Fundamentals
  – Sampling, Data sampling
  – Simple examples of analysis and importance, 1-D and 2-D
* Signal and Intensity Transformations
  – Intensity functions, 1-D and 2-D processing
  – Signal Transformation

4

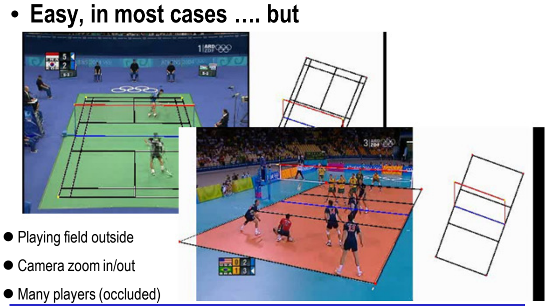
Module 01 – Part 1
Motivation for this course

Applications of sensing in sports & power of visual imaging

5

Application – (1) / Finding the playing field

* Easy, in most cases …. but

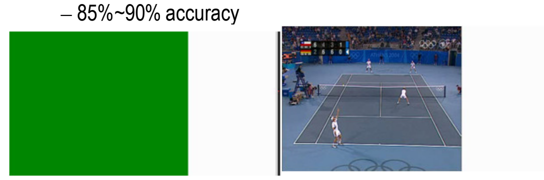


* Playing field outside
* Camera zoom in/out
* Many players (occluded)

6

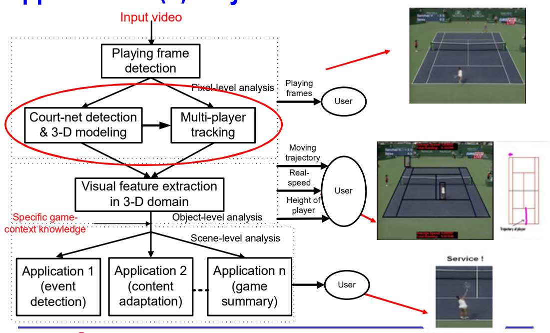
Application – (2) / Semantic analysis

* Application: semantic event detection
  – Analyze single match and output important events, like "service" and "net-approach"
  – Analyze double match and output basic tactics like "both back"
  – 85%~90% accuracy



Application – (3) / System overview: multi-level

7



Application – (4) / Sports Video Content Analysis

8

- **Motivation**
  - Large AV files (sports) are stored and retrieved
  - Users have interest in specific semantic parts
- **Overview related work (obj.tracking, & summarization**
  - Pixel/Object-level analysis: player tracking
  - Highlight detection: goal in football video
  - Shot-based event detection: player close-up, audience
- **Problem statement (missing points)**
  - Generic 3D model in system
  - Multi-level analysis results, i.s.o. semantic event
  - Near real-time operation

Personal training/fitness at home – (1) Use of personal devices

9

- Health/sport is growing to the home env.
- Enabling technology in consumer domain
- Off-the-shelf com-ponents (phone etc.)
- No medical validation except in prof. domain
- Data ownership is an issue

Personal training/fitness at home – (2) Use of personal devices

10

- Heart rate is a key parameter
- Temperature measurement
- Stress, effort, sleep motion, etc. also emerging
- Most parameters are 1-D signals
- Signals + analysis are both required!

Sensing of signals & required processing

11

- Previous examples are typical of sensing
- One-dimensional signals are the starting point
- Audio signals are also 1-D (or 2 channels), Images 2-D signals / Video 3-D (2-D + time) / 4-D (3-D + time)
- Typical processing steps

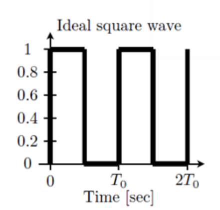
Sensor → Signal conversion → Signal processing → Signal analysis →

A/D conversion | De-noise, enhance | Find relevant information

Examples / 1-D signals & processing – (1)

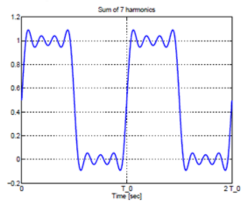
12

- Many 1-D signals are periodic
- Describe the signal / measure the properties
- Example properties are:
  - Minimum / maximum value
  - Periodicity / frequency
  - Contrast
  - Duty cycle
  - Other properties: quality, noise



Examples / 1-D signals & processing – (2)

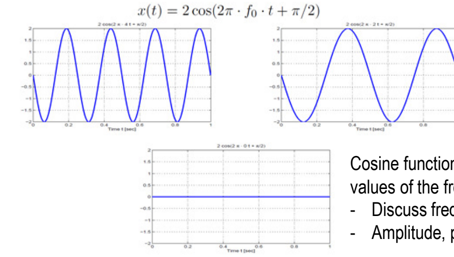
- Signal in 1-D is also periodic
- Difference with previous slide
- Example properties are:
  - Non-ideal rise time
  - Harmonics in signal
  - Looks like noise superimposed
- Enhancement may be needed
- Selective filtering: remove harmonics
- How to separate small/ large signals?



Examples / 1-D signals & processing – (3)

Amplitude $A = 2$, Phase $\phi = \pi/2$, Frequency f_0 varies $4 \to 0$ [Hz]

$$x(t) = 2\cos(2\pi \cdot f_0 \cdot t + \pi/2)$$



Cosine function for several values of the frequency
- Discuss frequency f_0
- Amplitude, phase

Examples / 1-D signals & processing – (4)

What is the amplitude, frequency and phase of a sinusoid?

Mathematical formula continuous-time cosine:

$$x(t) = A\cos(\omega_0 \cdot t + \phi) = A\cos(2\pi \cdot f_0 \cdot t + \phi)$$

Symbol	Name	Dimension
A	Amplitude	-
ω_0	Radian frequency	rad/sec
ϕ	phase	rad
f_0	(cyclic) frequency	$\sec^{-1}$ = Hz

Relation period - frequency:

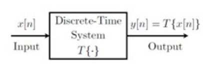
$$x(t + T_0) = x(t)$$

$$A\cos(\omega_0 \cdot (t + T_0) + \phi) = A\cos(\omega_0 \cdot t + \phi)$$

$$\omega_0 T_0 = 2\pi \;\Rightarrow\; T_0 = \frac{2\pi}{\omega_0} \quad \text{or} \quad (2\pi f_0)T_0 = 2\pi \;\Rightarrow\; T_0 = \frac{1}{f_0}$$

1-D Signals & Filtering – (1)

A filter is a system designed to remove some components or modify some characteristics of a signal.
Finite Impulse Response (FIR) filters: Each output sample is sum of finite number of weighted samples of input sequence



Both input and output are discrete-time samples (in contrast to e.g. D-to-C and C-to-D) $\to$ Operator $T\{\cdot\}$ described by formula

Examples:

$$y[n] = x^2[n]$$

$$y[n] = \max\{x[n], x[n-1], x[n-2]\}$$

1-D Signals & Filtering – (2)

Compute a moving (running) average of two or more consecutive samples, forming a new sequence of the average values

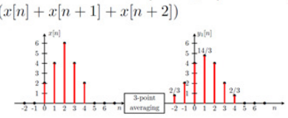
Note: FIR is generalization of running average

Averaging used when data fluctuates, thus smoothing prior interpretation (e.g. view trends). E.g. Stock-market prices, credit-card balances, etc.

Example:

Input-output relation (=difference equation) 3-point averaging

$\to y_1[n] = \frac{1}{3}\left(x[n] + x[n+1] + x[n+2]\right)$



1-D Signals & Filtering – (3)

Difference equation general FIR: $y[n] = \sum_{k=0}^{M} b_k x[n-k]$

- Weighted running average of $M+1$ samples
- Causal filter
- M is order of FIR filter
- $L = M + 1$ is length of FIR filter

Example: FIR of order $M = 3$ with coefficients $b_k = \{3, -1, 2, 1\}$

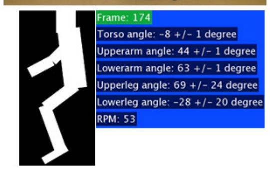
$$\to \quad y[n] = 3 \cdot x[n] - 1 \cdot x[n-1] + 2 \cdot x[n-2] + 1 \cdot x[n-3]$$

n	$n<0$	0	1	2	3	4	5	6	7	8	$n>8$
$x[n]$	0	2	4	6	4	2	0	0	0	0	0
$y[n]$	0	6	10	18	?	?	?	8	2	0	0

1-D Signals & Applications – (1)

19

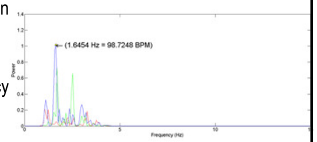
- **Video Training Analysis**
  - Automatically estimate the cyclist' pose and movements based on th video from a single camera.
- **Observation**
  - Feet motion follows sine wave (varying amplitude over time
- **Solution**
  - Foreground segmentation
  - Body part detection, pose estimate
  - Goal -> Measure cycling speed



1-D Signals & Applications – (2)

20

- **Goal**
  - **Automatic heart rate detection** based on the video frames
- **Sensing techniques**
  - Unobtrusive via video camera
  - Or use watch
- **Solution**
  - Temporal color variation extraction on forehead
  - Bandpass filtering
  - Peak detection in frequency domain



Reflection / Sports Technology Aspects – (1)

21

- **Motivation**
  - Technology is often used in sports for measuring performance
  - Health conditions of sports athlete
- **Technical solutions are based on**
  - Sensing: heart rate, ECG, Audio, Video
  - Digital Signal Processing: Video/Image, Audio, Other signals
  - Advanced Signal Analysis: Find heart rate, track the players, etc.
  - Modeling of events/data: Playing field, temporal behaviour, events

Reflection / Sports Technology Aspects – (2)

22

- **Generalizing on signals**
  - Learn general concept: video, audio, heart rate, ECG,… are **signals!**
  - Video is the most informative signal (samples, speed, bandwidth)
  - Many concepts in this course are re-applicable to other signals

23

Module 01 – Part 2
Image Fundamentals

Image sampling, quantization, representation, scaling & mathematics

Image sensing and acquisition

24

Line sensor

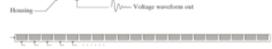
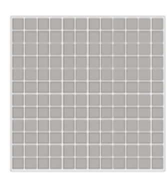


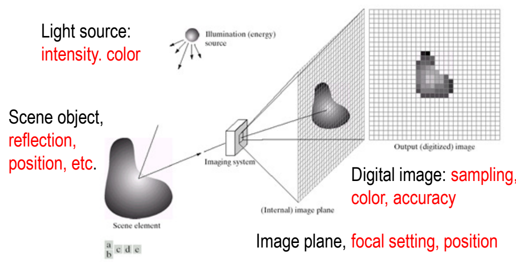
Image sensor:

- array of elements
- in picture/video cameras (CCD),
- X-ray equipment



Sensor is a converter from photoeletronic converter or radiation converter to electrical signal (key parameter is resolution)

Image formation model – (1)



a b c d e
FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Image formation model – (2)

Images are a 2-D function of the $f(x, y)$

∗ Value is a positive scalar function at (x, y)
- E.g. $0<f(x, y)<$ *infinite*
- Function f may be characterized by two components, the amount of illumination and reflection,

  thus $f(x, y) = i(x, y)\, r(x, y)$
- Where $0<i(x, y)<$ *infinite* and $0<r(x, y)< 1$

  (reflectance between total absorption and full reflectance)

Image sampling and quantization – (1)

- **Samples**: the time-discrete values of an electrical signal after A/D conversion
- **Pixels**: the full-color elements of a video picture or still image, e.g. RGB triplets
- **Line**: one row of pixels or samples from a (video) image
- **Image/Frame**: an individual image array of samples (can be from a sequence)
- Images can be *continuous* or *digitized*

Image sampling and quantization – (2)

- Digitization or discretization is a process implemented in two ways:
  - **Sampling**: the time-discrete values of function $f(t)$, giving $f(i)$ or $f(i,j)$, leading to image samples
  - **Quantization:** the amplitude-discrete output values of function $f(t)$ giving $f_d(t)$

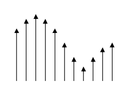
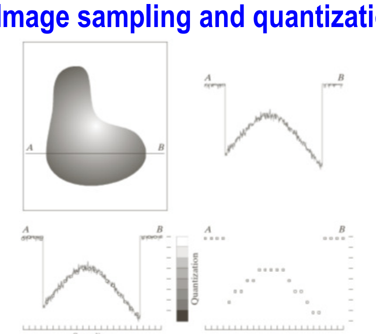


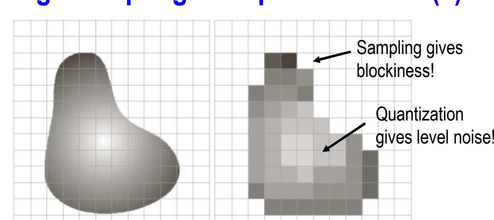
Image sampling and quantization – (3)



a b
c d

FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

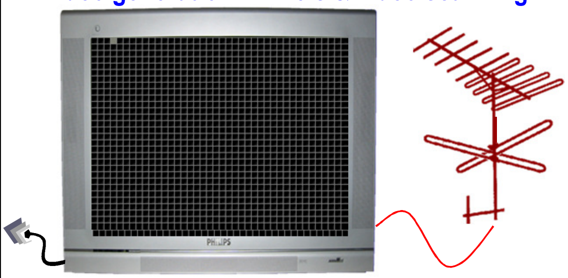
Image sampling and quantization – (4)



Sampling and Quantization introduce errors giving visual artifacts

a b
FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

Example / Video sampling in TV: Video generation – Pixels & video scanning



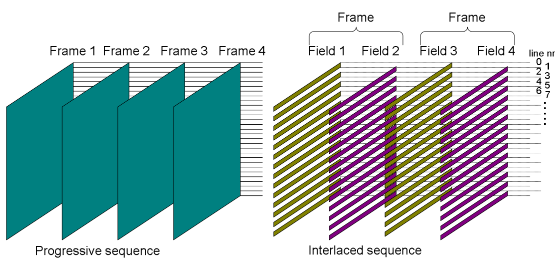
Note the differences between samples, pixels, lines, image and display!

Definitions & parameters of video signal

Characterization of a video signal

- 1. **Spatial Resolution** (the intrinsical sharpness…)
  - **SDTV** 720 x 576 (PAL in EU), 720 x 480 (NTSC (USA & JP)
  - **HDTV**: 1920 x 1080 ("full HD"), 1920 x 1080, 1344 x 720 ("HD ready"), etc.
- 2. **Temporal resolution** (frames rate)
  - EU: **25 Hz**, USA: **30 Hz**
- 3. Video **line scanning** process
  - **Progressive**: 1:1, each next line comes down the previous,
  - **Interlaced**: 2:1, one field with odd lines, and one field with even lines with a temporal difference

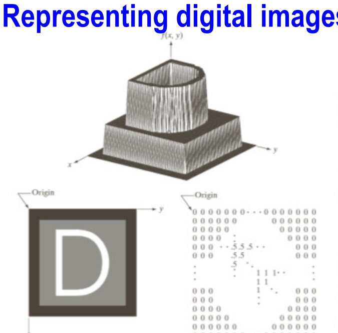
Progressive vs. interlaced video



Representing digital images – (1)

- **Assume $f(s, t)$ as a continuous image and let it be sampled and quantized into a 2-D array $f(x, y)$**
  - **Sampling**: $f(x, y)$ has M rows and N columns
  - $x = 0, 1, 2, \ldots, M-1$ and $y = 0, 1, 2, \ldots, N-1$
  - the value $f(0, 0)$ is at the origin
  - **Spatial domain**: the section of the real plane spanned by the coordinates of the image
  - x and y are the spatial variables/coordinates

Representing digital images – (2)



a
b c

FIGURE 2.18 (a) Image plotted as a surface. (b) Image displayed as a visual intensity array. (c) Image shown as a 2-D numerical array (0, .5, and 1 represent black, gray, and white, respectively).

Representing digital images – (3)

It is common to represent a digital image as a numerical array

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \ldots & f(0,N-1) \\ f(1,0) & f(1,1) & \ldots & f(1,N-1) \\ \ldots & \ldots & \ldots & \ldots \\ f(M-1,0) & f(M-1,1) & \ldots & f(M-1,N-1) \end{bmatrix}$$

Remarks

- Mind the notation of rows and columns!
- Alternatively, notation with subscripts is sometimes used
- One line out of the array leads to a 1-D signal $f(x)$

Representing digital images – (4)

Digitization in amplitudes

- **typically, the number of levels is set to a power of 2,**

$$L = 2^k$$

- **so the amplitudes or intensity are in** $[0, L\text{-}1]$
- **This interval** $[0, L\text{-}1]$ **is called dynamic range.**
  - **In practice, the upper level is called *saturation* and the lower level is limited to the *noise* level.**
- **Contrast is defined as the difference in intensity between the highest and the lowest level.**
- **The number of bits for the total image is now**

$$b = M \times N \times k$$

Representing digital images – (5)

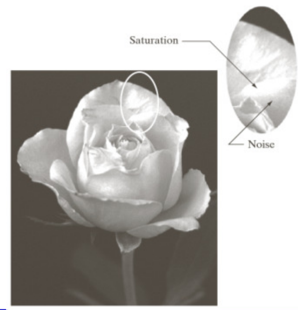


FIGURE 2.19 An image exhibiting saturation and noise. Saturation is the highest value beyond which all intensity levels are clipped (note how the entire saturated area has a high, *constant* intensity level). Noise in this case appears as a grainy texture pattern. Noise, especially in the darker regions of an image (e.g., the stem of the rose) masks the lowest detectable true intensity level.

Representing digital images – (6)

The bit costs for SD, HD etc images are significant, and in the medical case, bits/sample are between k = 8 – 14 bits!

TABLE 2.1
Number of storage bits for various values of N and k.

N/k	1 (L = 2)	2 (L = 4)	3 (L = 8)	4 (L = 16)	5 (L = 32)	6 (L = 64)	7 (L = 128)	8 (L = 256)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

Spatial and intensity resolution – (1)

Spatial resolution

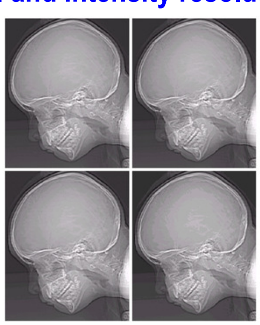
- **can be expressed in several ways**
  - **line pairs per unit distance (black & white line pairs in W, then pair width is $2W$, $1/2W$ pairs/unit dist)**
  - **dots per unit distance (printing dots per inch dpi)**
  - **typically in the number of samples N x M**

Intensity resolution

- **typically expressed in the number of bits / sample k**
- **Remember that** $b = M \times N \times k$

Spatial and intensity resolution – (2)

Variation of intensity resolution in a CT image



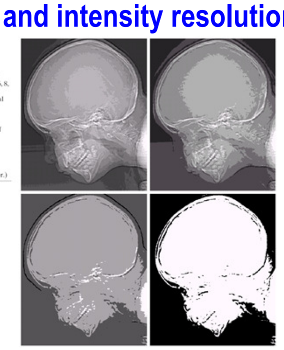
a b
c d

FIGURE 2.21 (a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

Spatial and intensity resolution – (3)

e f
g h

FIGURE 2.21 *(Continued)* (e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)



Variation of intensity resolution in a CT image giving false contouring

Spatial and intensity resolution – (4)

Variation of spatial resolution can give aliasing artifacts

Upscaling is performed with interpolation

a b c
d e f

FIGURE 2.24 (a) Image reduced to 72 dpi and zoomed back to its original size (3692 × 2812 pixels) using nearest neighbor interpolation. This figure is the same as Fig. 2.20(d). (b) Image shrunk and zoomed using bilinear interpolation. (c) Same as (b) but using bicubic interpolation. (d)–(f) Same sequence, but shrinking down to 150 dpi instead of 72 dpi [Fig. 2.24(d) is the same as Fig. 2.20(c)]. Compare Figs. 2.24(e) and (f), especially the latter, with the original image in Fig. 2.20(a).

Spatial resolution / interpolation – (1)

Spatial resolution: upconversion / interpolation

- interpolation is used after shrinking the image and then converting it back to the original size
- Interpolation is using the known data to estimate values at unknown locations
  - **example: interpolate 500x500 image to 750x750. Imagine *zooming* the image and then shrinking it to the original size.**
- **Nearest neighbor interpolation:** assign the intensity of the nearest neighbor from the original to the unknown location
- **Bilinear interpolation:** use the intensity of the 4 nearest neighbors to compute the unknown location:
  - $v(x, y) = ax + by + cxy + d$

•Note: a,b,c,d are coefficients from the 4-fold equation system

Spatial resolution / interpolation – (2)

Interpolation techniques

- **Bicubic interpolation:** computation based on the 16 nearest neighbors of a sample point, using the relation

$$v(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j$$

- where a 16-fold equation system has to be solved for coeff. a_{ij}.
- HV Neighbors of a sample: $(x-1, y)$, $(x+1, y)$, $(x, y-1)$, $(x, y+1)$

Perceptive quality differences

- Pixel subsitution generates aliasing effects on edges
- Bicubic preserves better detail than bilinear interpolation
- Bilinear interpolation is not a linear function!

Array and matrix operations – (1)

Array product

- **Array product** relates to the product of two sub-images

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

- Compare this to the matrix product! (Having the extra terms due to row-column expansion of the product)

Linear operations – (1)

Linear operator

- A general operator H producing an output $g(x, y)$ for input image $f(x, y)$ satisfying $H[f(x, y)] = g(x, y)$ is said to be a linear operator if

$$H[a_i f_i(x, y) + a_j f_j(x, y)] = a_i H[f_i(x, y)] + a_j H[f_j(x, y)]$$

•The first underlying property is called additivity, saying that the output of the operator on two sums is equal to the sum of the individual operations

- A second property is of homogeneity, which says that

$$H[a_i f_i(x, y)] = a_i H[f_i(x, y)]$$

Non-Linear operations

***Max* is a non-linear operator**

- Verify and prove that the non-linear operator *max* applied in

$$H[a_i f_i(x, y) + a_j f_j(x, y)] = a_i H[f_i(x, y)] + a_j H[f_j(x, y)]$$

with

$$f_1(x, y) = \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}, f_2(x, y) = \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}, a_1 = 1, a_2 = -1$$

gives at the left side -2 and at the right-hand side -4.

Both linear and non-linear operators are used in image processing!

Arithmetic operations – (1)

Arithmetic operations between images are array operations, meaning that the arithmetic operations are applied to the corresponding pixel pairs.

• the four arithmetic operations are

$$s(x,y) = f(x,y) + g(x,y)$$
$$d(x,y) = f(x,y) - g(x,y)$$
$$p(x,y) = f(x,y) \times g(x,y)$$
$$v(x,y) = f(x,y) \div g(x,y)$$

Where f and g are images of M rows and N columns, and the resulting images also. (example: s is an image signal with noise added)

Arithmetic operations – (2) - Examples

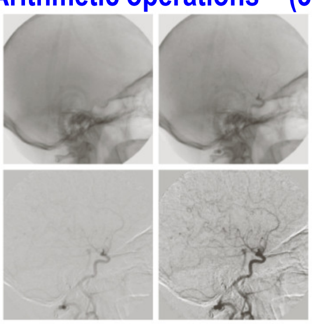
Arithmetic operations: DSA subtraction between images

• In Digital Subtraction Angiography, images are subtracted before and after injecting an X-ray contrast medium in the blood vessels of the same patient body area.

$$d(x,y) = f(x,y) - g(x,y)$$

• Image segmentation. Here a common technique is to subtract the static background without objects from the usual scene with typical moving foreground objects in it. The subtraction gives the foreground objects, that can be further filtered and clustered for clarity.

Arithmetic operations – (3) - DSA



a b
c d

FIGURE 2.28 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

Module 01 – Part 3
Intensity transformation and filtering

Filtering basics, Log transform, histograms, spatial filtering and types

Basics of filtering and intensity – (1)

• A standard filtering action is performed in sample (spatial) domain
• Spatial processing is typically less complex in computations and require less resources
• The choice for spatial or transform domain processing depends on (a) the problem and (b) allowed costs.
• Spatial domain processing applies to

$$g(x,y) = T[f(x,y)]$$

• Operator T works on input f and has a neighborhood of point (x, y).
• Examples: pixel-by-pixel sum over set of images for noise reduction

Basics of filtering and intensity – (2)

• A standard filtering technique looks as follows: apply processing with neigborhood and move central pixel to the right (pixel-by-pixel proc.)

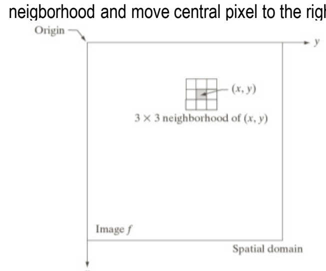
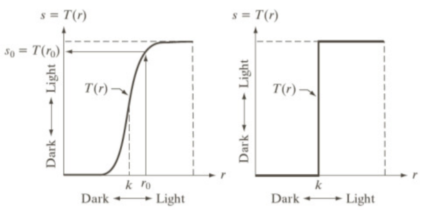


FIGURE 3.1 A 3 × 3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

Basics of filtering and intensity – (3)

- The smallest neighborhood is 1x1 pixel: the pixel itself only.
- Applying a function $T[\,f(x,\ y)\,]$ means transforming each pixel with that function. This gives a new pixel $g(x,\ y)$, again pixel-by-pixel proc.



a b

FIGURE 3.2 Intensity transformation functions. (a) Contrast-stretching function. (b) Thresholding function.

Contrast stretching Contrast clipping: thresholding

Intensity transformations – (1) / Negative

- Applying a function $s = T[\ r\ ]$ transforms a pixel into another value.
- Example1: **image negative**, apply function s=L-1-r to range [0, L-1]
- Example 2: log function, apply the function s=$c$log(1+r) with r>=0. This function 'spreads' intensities, compress the **dynamic range**

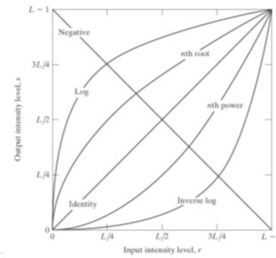
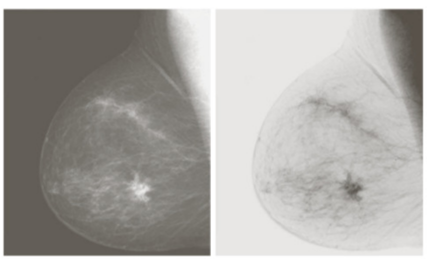


FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

Intensity transformations – (2) / Negative

Example illustration of the image negative function, a mammogram.



a b

FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

Intensity transformations – (3) / Gamma

Power-law (Gamma) transformations, function $s = c\ r^{\gamma}$, with $c, \gamma > 0$

- Functions with γ>1 and γ <1 have the opposite effect!
- The exponent is the gamma and the function is called gamma function

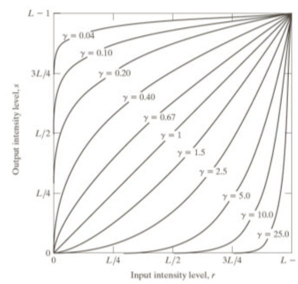
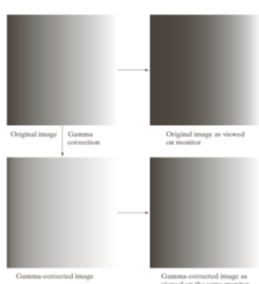


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

Intensity transformations – (4) / Gamma

Power-law (Gamma) transformations, function $s = c\ r^{\gamma}$, with $c, \gamma > 0$

- Gamma correction: I-V function in CRT has γ =2.2 (take 2.5).
- If a picture is shown, it tends to be too dark.
- With gamma correction the inverse function is applied at capturing (camera) so that the picture is already pre-corrected.

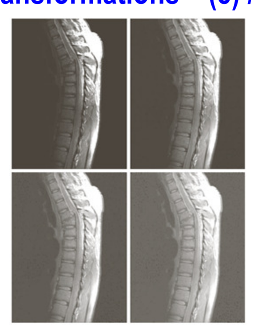


a b
c d

FIGURE 3.7 (a) Intensity ramp image. (b) Image as viewed on a simulated monitor with a gamma of 2.5. (c) Gamma-corrected image. (d) Corrected image as viewed on the same monitor. Compare (d) and (a).

Intensity transformations – (5) / Gamma

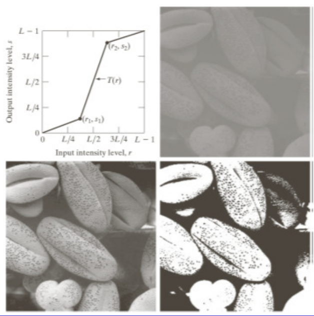
Gamma correction is useful in e.g. correcting MRI images, as is shown in visualizing an upper thoracic human spine.



a b
c d

FIGURE 3.8 (a) Magnetic resonance image (MRI) of a fractured human spine. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6$, 0.4, and 0.3, respectively. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Intensity transformations – (6) / Contrast

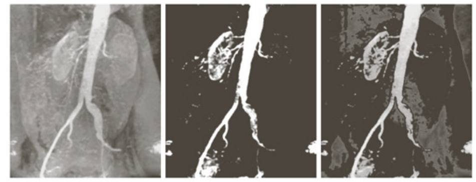


a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Intensity transformat's – (7) / Bit slicing

With bit slicing, we select a subset of the bit layers or a specific area of the bit gray scale (e.g. here lift up gray scale to white) .



a b c

FIGURE 3.12 (a) Aortic angiogram. (b) Result of using a slicing transformation of the type illustrated in Fig. 3.11(a), with the range of intensities of interest selected in the upper end of the gray scale. (c) Result of using the transformation in Fig. 3.11(b), with the selected area set to black, so that grays in the area of the blood vessels and kidneys were preserved. (Original image courtesy of Dr. Thomas R. Gest, University of Michigan Medical School.)

Histogram processing – (1)

- Histogram of image is a discrete function $h(r_k) = n_k$, where n_k is the number of intensity levels equal to r_k.
- Common is to normalize of the number of image samples MN.
- Then the normalized histogram becomes $p(r_k) = n_k / MN$.
- Histogram functions are used many times.

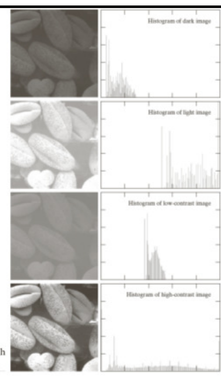


FIGURE 3.16 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

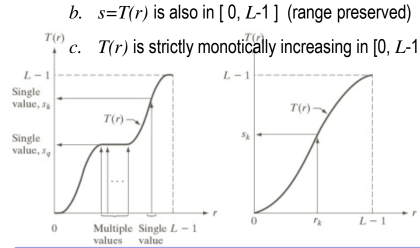
Histogram processing – (2) / Equalization

Consider images with (contin.) intensities r in [0, L-1] and notice mappings of the form $s = T(r)$, where

a. $T(r)$ is monotonically increasing (no inversions)

b. $s=T(r)$ is also in [0, L-1] (range preserved)

c. $T(r)$ is strictly monotically increasing in [0, L-1]



a b

FIGURE 3.17 (a) Monotonically increasing function, showing how multiple values can map to a single value. (b) Strictly monotonically increasing function. This is a one-to-one mapping, both ways.

Histogram processing – (3) / Equalization

Consider images with (contin.) intensities r in [0, L-1] and note that the intensity levels may be random variables in the interval.

If the PDF of r is known and $T(r)$ is a continuous differentiable function

a. $p_s(s) = p_r(r)\,|dr / ds|$

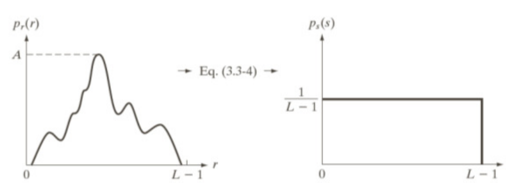
b. $s = T(r) = (L-1)\int_0^r p_r(w)dw$

c. Right hand side is the cumulative PDF and we can verify that both previous slide conditions a) and b) hold

d. Then $\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)dw\right] = (L-1)p_r(r)$

e. Substituting this, gives $p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = \frac{1}{L-1}$

Histogram processing – (4) / Equalization



a b

FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

67

Histogram processing – (5) / Equalization

In the discrete case, integrals become summations and

- $p_r(r_k) = \frac{n_k}{MN}$ for k =0, 1, 2, …, L-1
- plotting this probability for values of r_k gives a **histogram**
- Discrete integration over r gives

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j$$

- Thus mapping each intensity of the pixels to a new image s with this integration is called **histogram equalization**

68

Histogram – (6) / Equalization example

In a 3-bit image (L=8) of size 64x64=4096 the original histogram is [0.19, 0.25, 0.21, 0.16, 0.08, 0.06, 0.03, 0.02]

- $p_r(r_k) = \frac{n_k}{MN}$ for k =0, 1, 2, …, L-1
- Discrete integration over r gives

$$s_0 = T(r_0) = (8-1)\sum_{j=0}^{k=0} p_r(r_j) = 7p_r(r_0) = 1.33$$

$$s_1 = T(r_1) = (8-1)\sum_{j=0}^{k=1} p_r(r_j) = 7(p_r(r_0) + p_r(r_1)) = 3.08$$

- Then values of s may be rounded to their nearest integer and normalization is required

69

Histogram – (7) / Equalization example

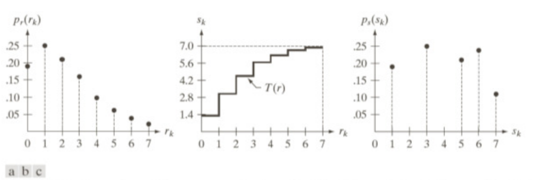


FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

- Histogram is an approximation to a PDF!
- Discrete histogram equalization does not yield a flat uniform distribution

70

Histogram – (8) / Equalization results

Results with

a. Too dark and low contrast
b. Too bright and low contr.
c. Low contrast and gray
d. Similar contrast

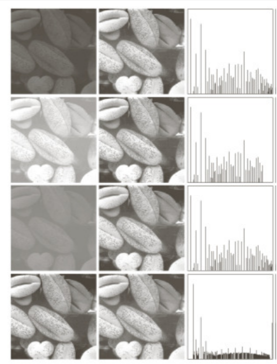


FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

71

Spatial filtering – (1) / Basics

Filtering is applied in (1) a **neighborhood** (typ. small rectangle) with (2) a **predefined operation**.

- If the operation is linear (multiply / add) it is **linear filtering**
- Assume a window or mask of mxn with m and n typically odd hence m=2a+1 and n=2b+1, then filtering is equal to

$$g(x,y) = \sum_{s=-a}^{+a}\sum_{t=-b}^{+b} w(s,t) f(x+s, y+t)$$

- Each pixel in the image is addressed with the mask, so x and y are varied. This called also **convolution** of w and f.

72

Spatial filtering – (2) / Basics

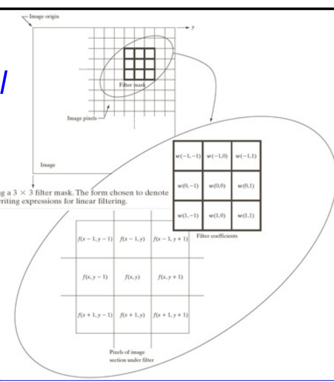


FIGURE 3.28 The mechanics of linear spatial filtering using a 3 × 3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

Spatial filtering – (3) / Basics

Numerical illustration of convolution and correlation

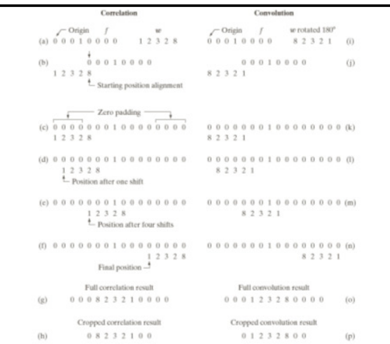


FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

Spatial filtering – (3) / Basics

Convolving a 2-D mask with a unit impulse creates the mask values in the image

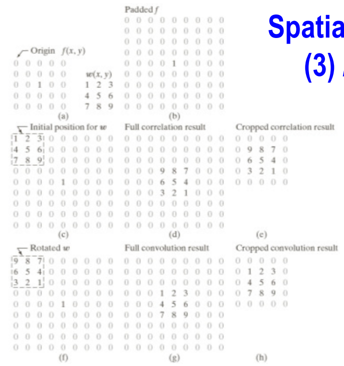


FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Spatial filters – (4) / Vector representation

Filtering can be rewritten into a **vector form**

- Sum of products is equal to the inner product of two vectors
- This gives

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn} =$$

$$R = \sum_{k=1}^{mn} w_k z_k = \mathbf{w}^T \mathbf{z}$$

- Convolution: rotate masks with 180 degrees, for correlation, use the mask given

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

FIGURE 3.31 Another representation of a general 3 × 3 filter mask.

Spatial filters – (5) / Smoothing filters

- Smoothing filters are **low-pass filters**, and take away details
- Sometimes also calles averaging filters (see example below)
- First example is called a box filter
- Second example is a **weighted average**

a b

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

FIGURE 3.32 Two 3 × 3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.

Spatial filters – (6) / Smoothing filters

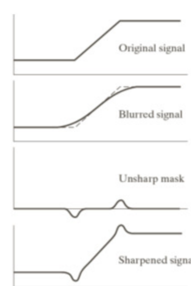
FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes m = 3, 5, 9, 15, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

a b
c d
e f

Spatial filters – (6) / Sharpening principle

With **unsharp masking**, the following steps are taken

1. Blur original image
2. Subtract the blurred image from original
3. Add the mask to the signal



a b c d

FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

Spatial filters – (7) / Sharpening filters

- Sharpening filters are often based on **derivatives of the signal**
- Taking **first derivatives** of a function must give that
  - (1) 1st derivative is zero when intensity is constant
  - (2) 1st derivative is non-zero at the transition of the signal or ramps
  - (3) 1st derivative is non-zero along signal ramps
- Similarly, the **second derivative** takes the following conditions
  - (1) 2nd derivative is zero in flat areas
  - (2) 2nd derivative is non-zero at the onset and end of a step
  - (3) 2nd derivative is zero along ramps and rises of fixed slope

Spatial filters – (8) / Sharp. 1-D derivation

One-dimensional derivation of a discrete signal *f(x)*

- Sharpening filters are often based on **derivatives of the signal**
- Taking **first derivatives** of a function is performed with

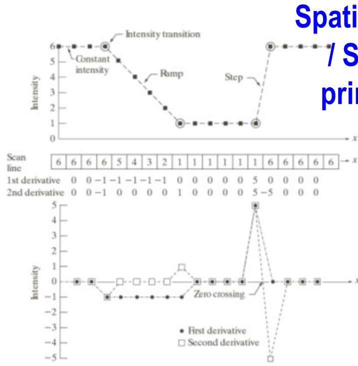
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- Similarly, the **second derivative** takes the following conditions

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

- Ex: Which hidden filter coefficients do you find?

Spatial filters – (9) / Sharpening principles 1-D



a b c

FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

Spatial filters – (10) / Sharp. Deriv. in 2-D

- Laplacian filters are based on the 2nd **derivatives of the signal**
- Laplacian filters are also used in image segmentation
- Interest in isotropic filters: response is independent of the direction of the signal discontinuities to which the filter is applied; e.g. a simple form is

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- which gives finally

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y), etc.$$

$$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Spatial filters – (11) / Sharpening 2-D filters

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

0	−1	0
−1	4	−1
0	−1	0

(d)

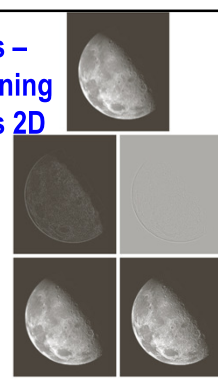
−1	−1	−1
−1	8	−1
−1	−1	−1

a b
c d

FIGURE 3.37 (a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

Spatial filters – (12) / Sharpening filters results 2D

Laplacian filters highlight discontinuities and edges and deemphasize slowly varying intensity areas



a
b c
d e

FIGURE 3.38 (a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)