Fundamental notions

Prof. Touradj Ebrahimi
Touradj.Ebrahimi@epfl.ch

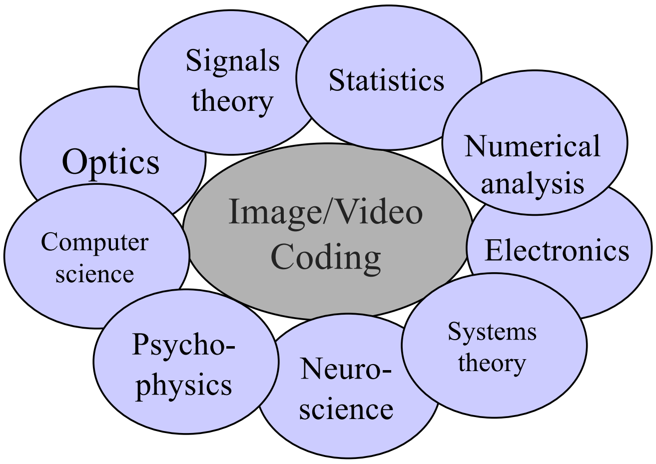
Image and video coding

- Examples of applications
  - Digital photography
  - Digital TV / HDTV / UHDTV / 3DTV
  - DVD / Blu-ray
  - VCR, PTR, PVR
  - Video surveillance
  - Medical imaging
  - Video conferencing
  - Video streaming
  - Multimedia-enabled smartphones
  - Multimedia Personal Computers
  - Computer / Machine vision in Robotics and IoT
  - Social media
  - Metaverse: VR, AR, MR
  - Synthetic image/video using Generative AI: Deepfakes
  - …

Image and video coding

3

- Relationship between image/video coding and other disciplines

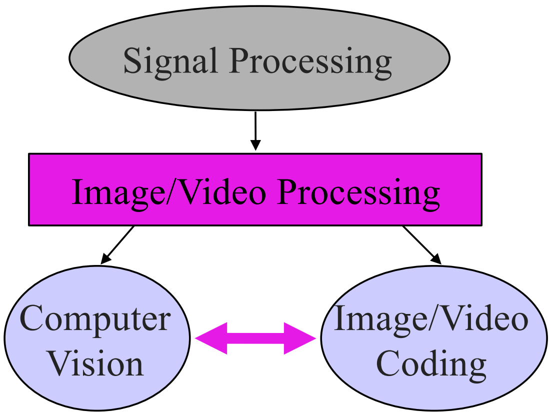


3

Image and video coding

4

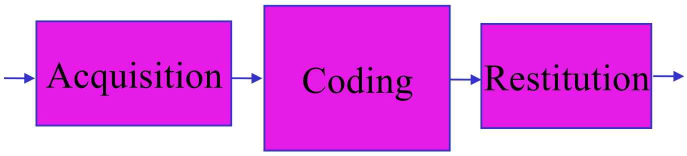
- Relationship between Signal Processing, Image/Video Processing, Image/Video Coding and Computer Vision



4

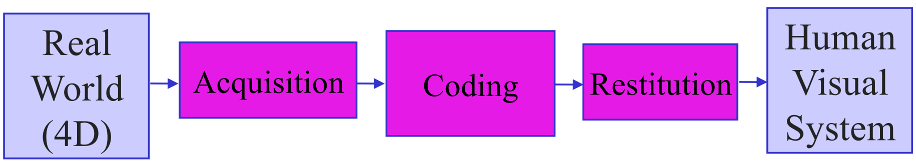
Systemics I/III

- Conventional chain of image/video coding



Systemics II/III

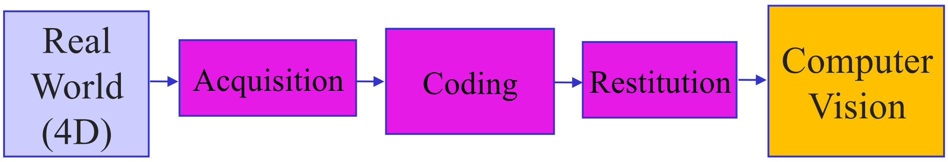
- Complete chain of image/video coding



Systemics III/III

7

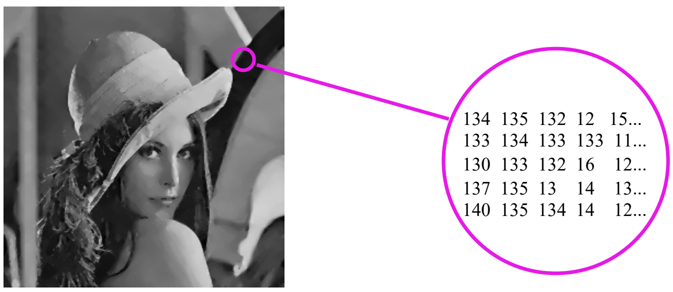
- Complete chain of image/video coding



7

Digital images

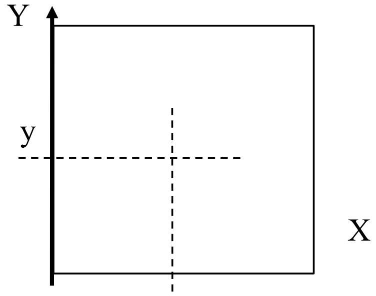
8



8

Canonical representation of a gray-level image

- An image is represented as a function $f(x,y)$ defined on a support of finite or infinite size. Variables x et y represent the spatial coordinates of a given point in the image, the value of the function (represented by a real number) defines the luminance (gray-level) associated with point (x,y)



Canonical representation of a gray-level image

A digital image $s(k,l)$ of size $K \times L$ is defined by a matrix of the same size:

$$s(k,l) = \begin{bmatrix} s(0,0) & s(0,1) & \cdots & s(0,L-2) & s(0,L-1) \\ s(1,0) & s(1,1) & \cdots & s(1,L-2) & s(1,L-1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s(K-2,0) & s(K-2,1) & \cdots & s(K-2,L-2) & s(K-2,L-1) \\ s(K-1,0) & s(K-1,0) & \cdots & s(K-1,L-2) & s(K-1,L-1) \end{bmatrix}$$

Every element of the matrix, also called a pixel, is represented by a limited number of bits

Image/video digitization

- Digitization is an essential step to go from a continuous (analog) to a discrete representation
- Two major components
  - Sampling
  - Quantization

Comparison with 1-D case

- In principle, all theoretical developments seen in 1-D case are also valid in M-D case by means of a simple generalization

- It is however often difficult, insufficient, and even dangerous to limit such developments to simple generalization of 1-D to M-D

Sampling

Sampling

- Definition
- Hypotheses
- Theory
- Practice
- Characteristics of sampled M-D signals

Sampling - Definition

- Periodic sampling of values of an analog signal
- Example for a 2-D signal

$$f(x,y)$$

$$s(k,l) \equiv s_e(x,y)\Big|_{x=k\Delta x, y=l\Delta y}$$

$$s_e(x,y) = \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} f(x,y)\delta(x-k\Delta x, y-l\Delta y)$$

Δx Sampling step for dimension X

Δy Sampling step for dimension Y

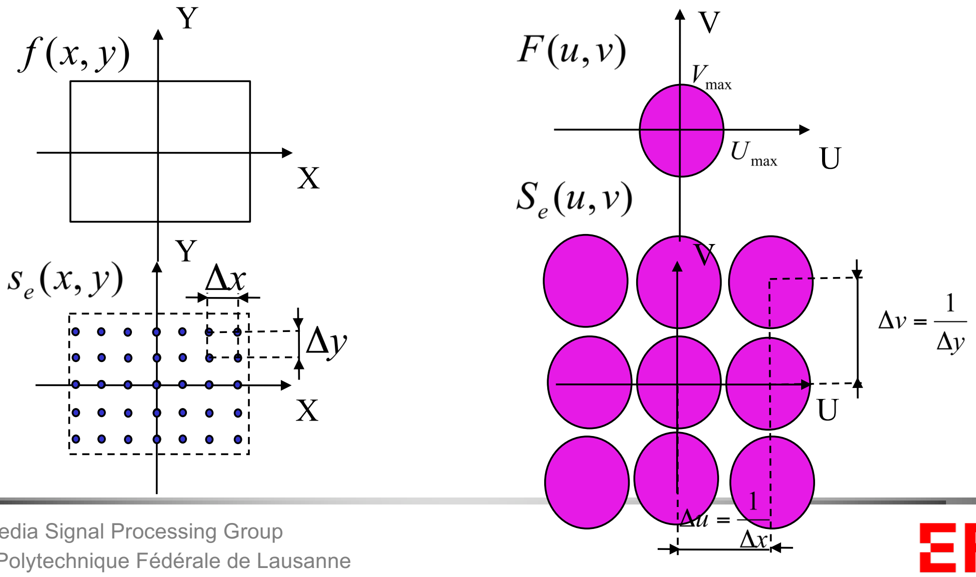
Sampling

- This relationship in the frequency domain becomes:

$$S_e(u,v) = \frac{1}{\Delta x \Delta y} \sum_{k=-\infty}^{+\infty} \sum_{l=-\infty}^{+\infty} F(u-\frac{k}{\Delta x}, v-\frac{l}{\Delta y})$$

Sampling – Graphical representation

- Spatial domain
- Frequency domain



Sampling theorem

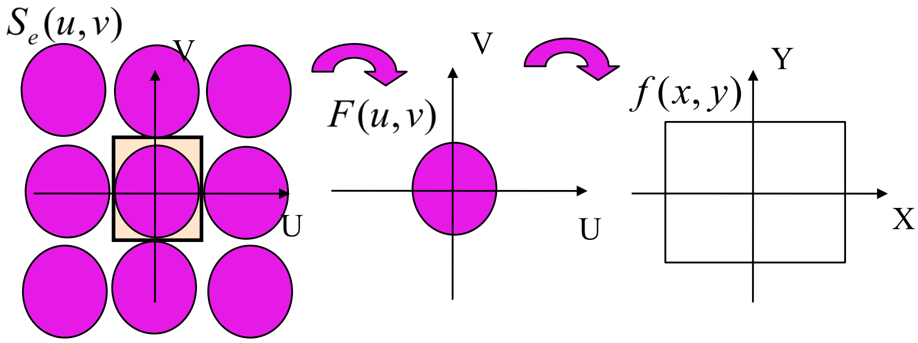
- An analog signal can be perfectly reconstructed from its samples as long as the sampling frequency is at least twice the amount of the maximum frequency component present in the analog signal

$$\frac{1}{\Delta x} \geq 2U_{max} \qquad \frac{1}{\Delta y} \geq 2V_{max}$$

Reconstruction

19

- In practice, reconstruction of an analog signal from its samples is performed by making use of a low-pass filter



Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

19

Hypotheses

20

- The maximum frequency component of the analog signal is known
- Signal is stationary

Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

20

Practice

21

- Sampling filter
- Oversampling
- Sampler low-pass filtering effect

$$s_e(x_0, y_0) = \iint f(\alpha, \beta) e(x_0 - \alpha, y_0 - \beta) d\alpha d\beta$$

$$S_e(u,v) = F(u,v)E(u,v)$$

- Optical filtering

Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

21

Sampling characteristics

22

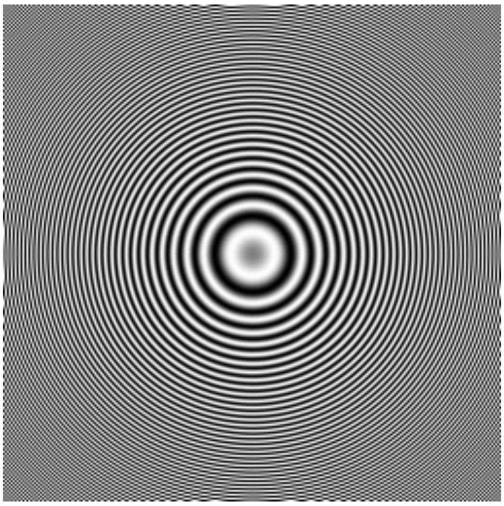
- A sampling without proper precaution can lead to spectral overlap
- Additional frequency components appear in the reconstructed signal
- Moiré patterns

Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

22

Moiré patterns

23



Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne

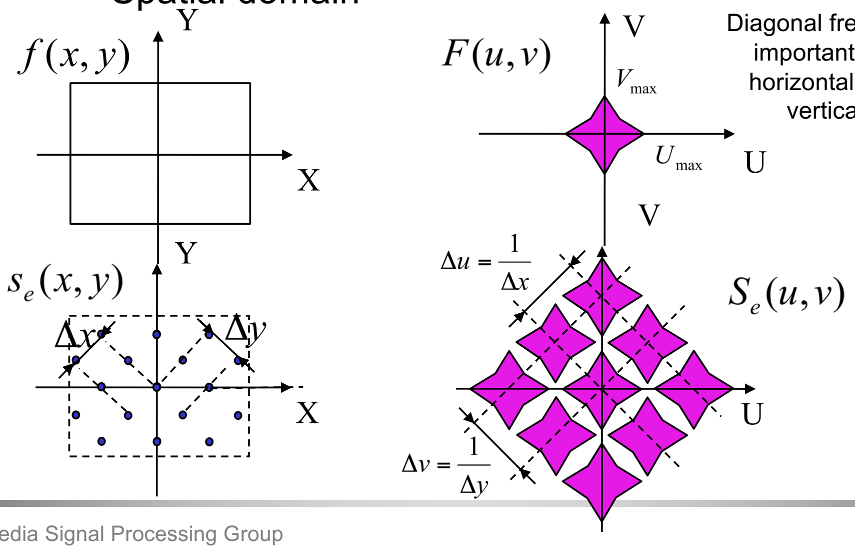
EPFL

23

Quincunx sampling

24

- Spatial domain
- Frequency domain



Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne

EPFL

24

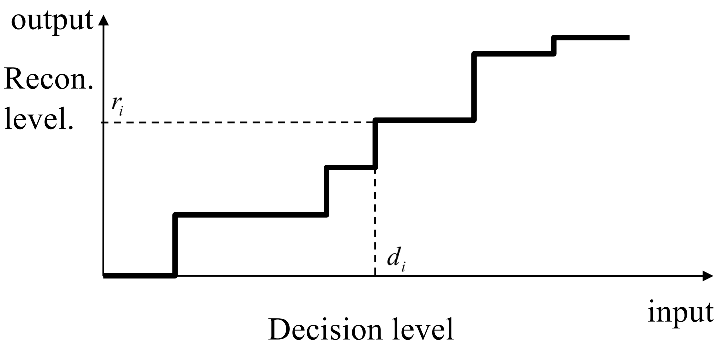
Quantization

Quantization

- Definitions
- Quantization noise
- Optimal quantization (Lloyd-Max)
- Uniform quantization
- Perceptual quantization
- Non-uniform quantization
- Compander
- Color quantization
- Vector quantization
  - see compression

Quantization - definition

- Projection of a signal with continuous amplitude into a set of finite number of discrete values



Quantization - definition

- General formulation

$$s_q = Q(s) \quad \text{such that } d_i \le s < d_{i+1} \Rightarrow s_q = r_i$$

- The problem of quantization consists in finding good values $\{d_i\}_i$ and $\{r_i\}_i$, as a function of the statistics of the original signal such that one can obtain the best approximation possible

Quantization noise

- Quantization noise

$$e = s - s_q = s - r_i$$

- Mean Square Error

$$\varepsilon = E\left[e^2\right] = E\left[(s - r_i)^2\right]$$

- If the probability density function of the signal is known:

$$\varepsilon = \int_D (s - r_i)^2 \, p_s(s) ds \qquad \int_D p_s(s) ds = 1$$

Optimal quantization

- For a pre-defined number of quantization levels Nq:

$$\varepsilon = \sum_{i=1}^{Nq} \int_{d_i}^{d_{i+1}} (s - r_i)^2 \, p_s(s) ds$$

- Optimal solution:

$$d_i = \frac{r_i + r_{i-1}}{2} \qquad r_i = \frac{\int_{di}^{di+1} s . p_s(s) ds}{\int_{di}^{di+1} p_s(s) ds}$$

Optimal quantization

- With a large number of dense quantization levels :

$$d_{i+1} \cong \frac{D \int_{d1}^{z_i+d_1} [p_s(s)]^{-1/3} ds}{\int_{d1}^{d_{Nq+1}} [p_s(s)]^{-1/3} ds} + d_1 \qquad D = d_{Nq+1} - d_1$$

$$z_k = (k / N_q).D$$

$$\varepsilon \cong \frac{1}{12 N_q^2} \left\{ \int_{d_1}^{d_{Nq+1}} [p_s(s)]^{1/3} ds \right\}^3$$

Uniform quantization

- Hypothesis : probability density function is uniform
- Q(s) is completely defined by a single and constant quantization step size

$$\Delta = \frac{d_{Nq+1} - d_1}{N_q} \qquad d_i = d_{i-1} + \Delta \qquad r_i = d_i + \Delta / 2$$

- Quantization error:

$$\varepsilon = \frac{\Delta^2}{12}$$

Weber law

33

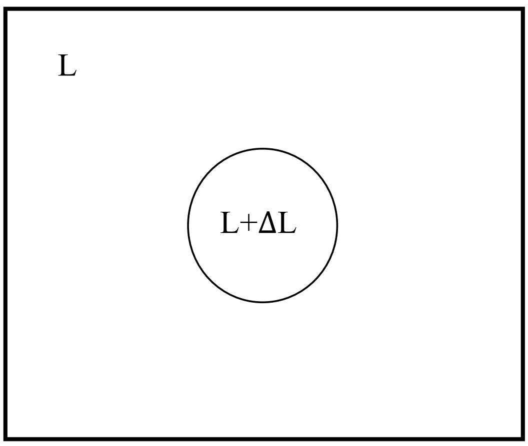
- Human eye is more sensitive to dark gray than light gray
- Weber-Fechner experiment
- Weber constant

Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

33

Weber experiment (1)

34



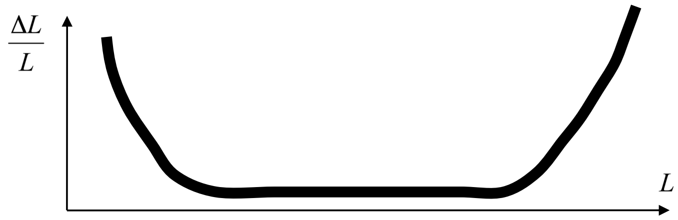
Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

34

Weber law

35

- Weber constant

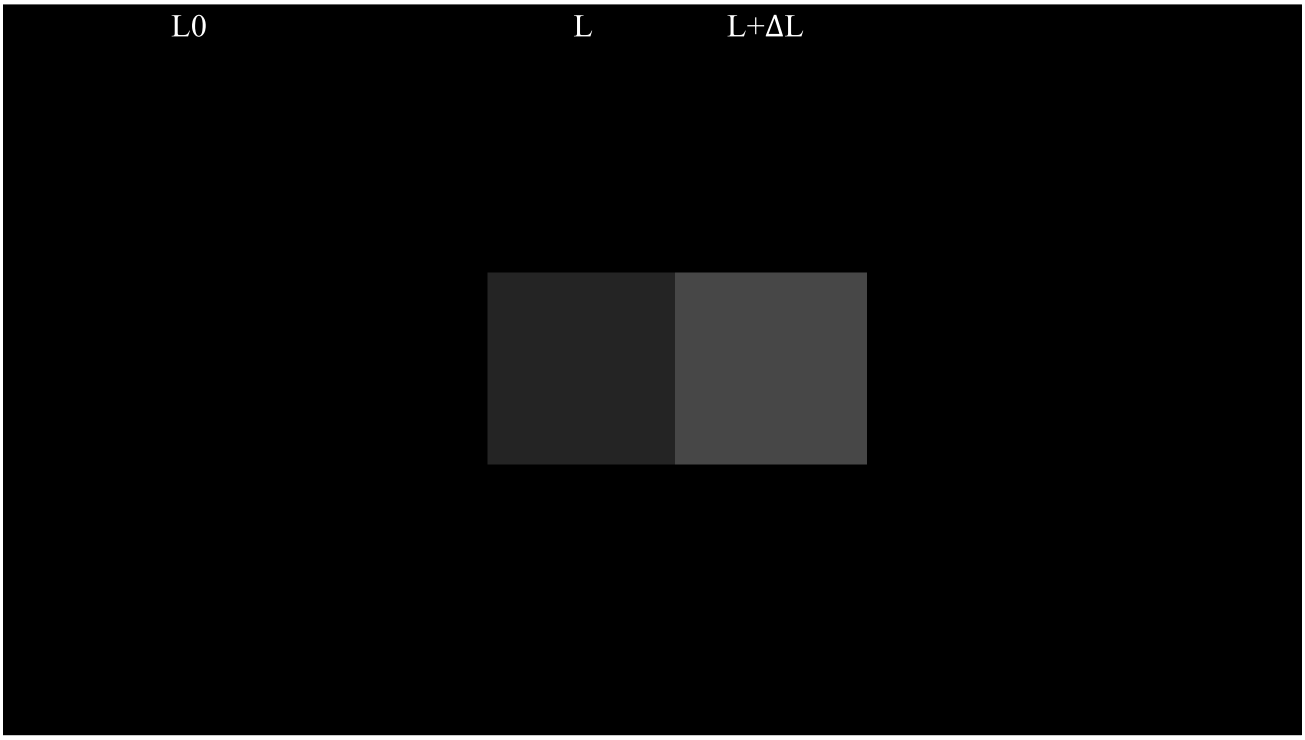


$$C_w = \frac{\Delta L}{L} \qquad C_w = 0.01 \ldots 0.02$$

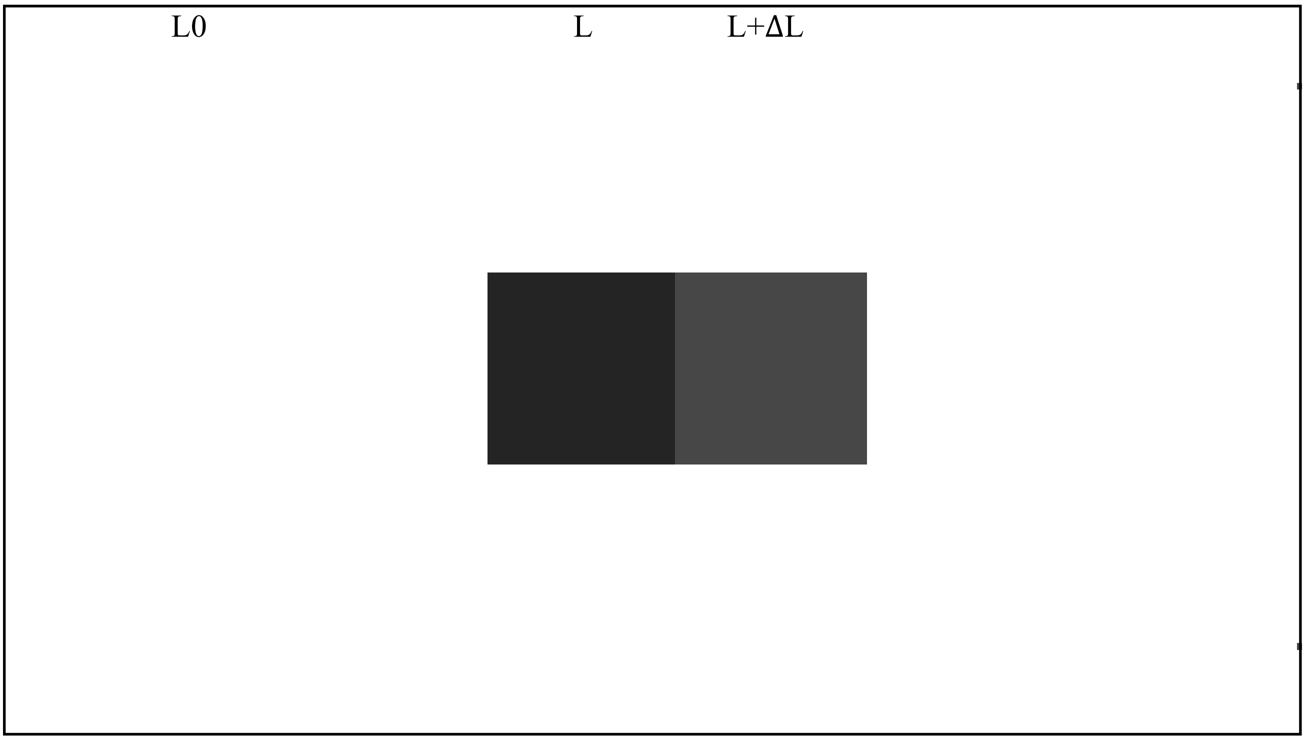
Weber constant

36

- Depends on many parameters:
  - Observer
  - Ambient lighting
  - Background luminance
  - Type of display
  - …



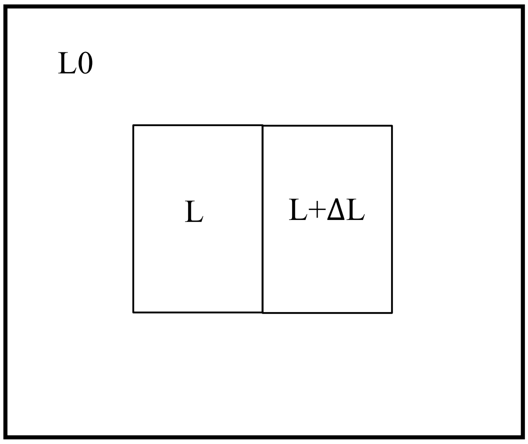
37



38

Weber experiment (2)

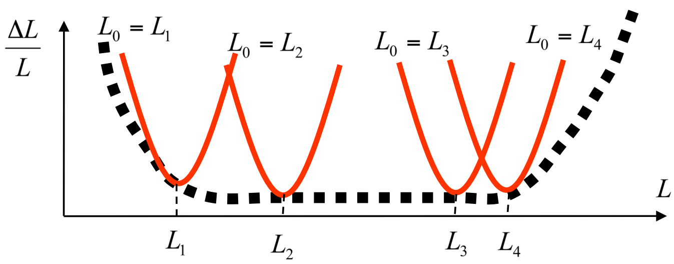
39



Weber law

40

- Weber constant



Choice of quantization step size

41

- Objective metrics to predict quantization distortions
  - MSE
  - PSNR
- Visual distortions due to over-quantization
  - false contours
- Pre- and post-processing methods to reduce false contours
  - image rendering

41

False contours

42

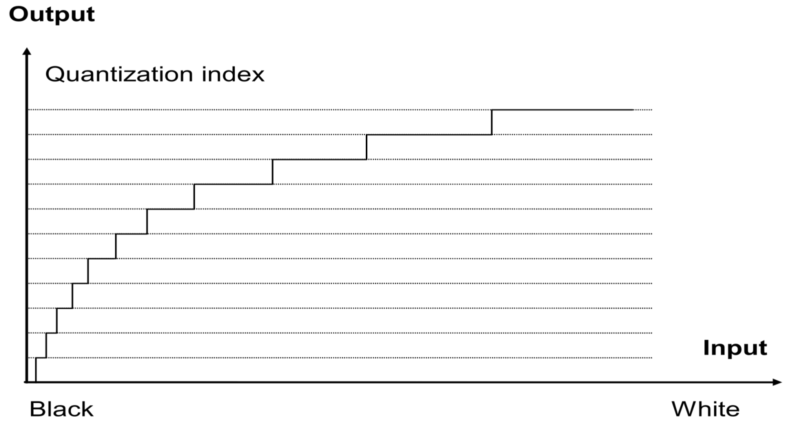
256
quantization levels

8
quantization levels

42

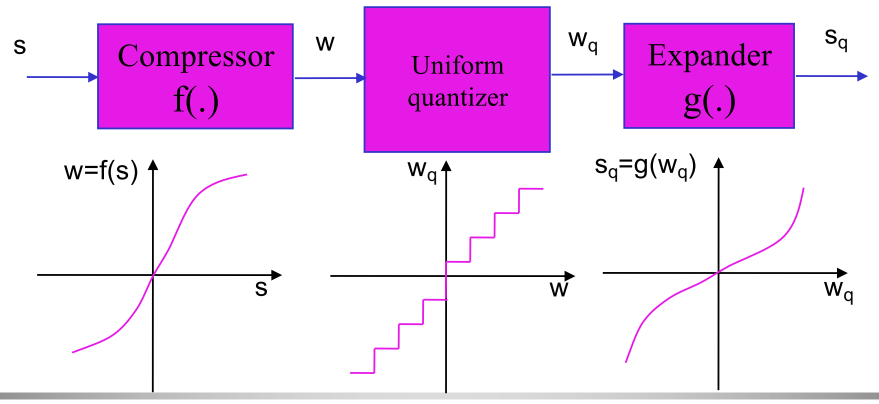
Non-uniform quantization

- When the probability density function of a signal is not uniform, non-uniform quantization becomes a better choice.
- Non-uniform quantization takes better into account the non-linear properties of the human visual system



Compander

- A compander is a uniform quantizer preceded and followed by non-linear transforms



Compander

- When $w \in [-a, a]$

$$f(x) = 2a \left\{ \frac{\int_{d_1}^{x} [p_s(s)]^{1/3} ds}{\int_{d_1}^{d_{Nq+1}} [p_s(s)]^{1/3} ds} \right\} - a$$

$$g(x) = f^{-1}(x)$$

Compander

- When $p_s(s) = p_s(-s)$

$$f(x) = a \left\{ \frac{\int_{0}^{x} [p_s(s)]^{1/3} ds}{\int_{0}^{d_{Nq+1}} [p_s(s)]^{1/3} ds} \right\} \text{ if } x \geq 0$$

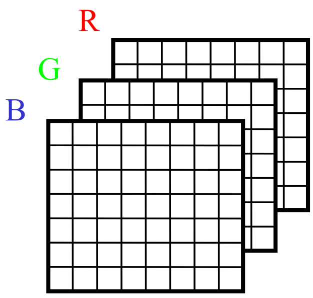
$$f(x) = -f(-x) \text{ if } x < 0$$

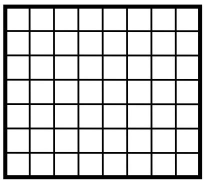
$$g(x) = f^{-1}(x)$$

Quantization of color images

- Each color component is quantized separately
- Some color components can be quantized (and even sampled) with different step sizes
- Alternative: Color quantization based on Look Up Tables (LUT)

Look Up Table (LUT)





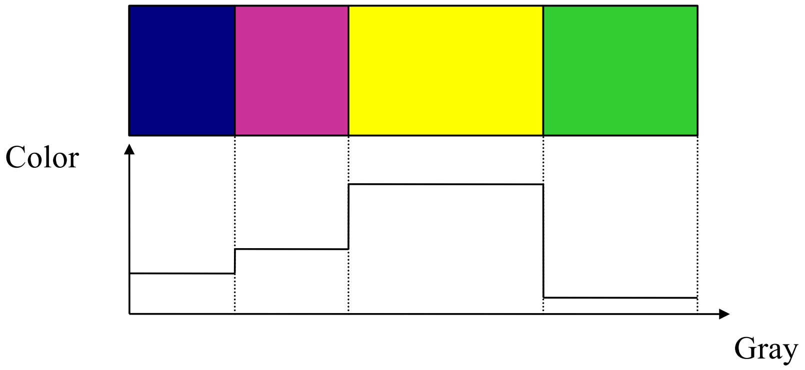
Valeur	R	G	B
0	10	10	10
1	10	20	30
2	30	100	20
...	...	...	...

True colors

Look Up Tables

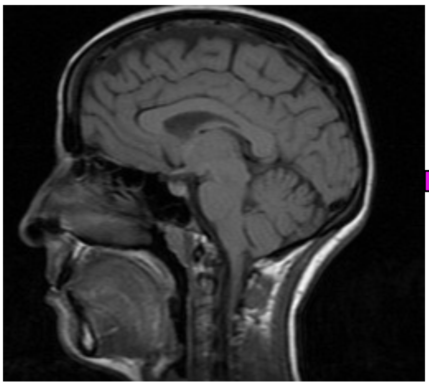
False colors

- Special LUTs

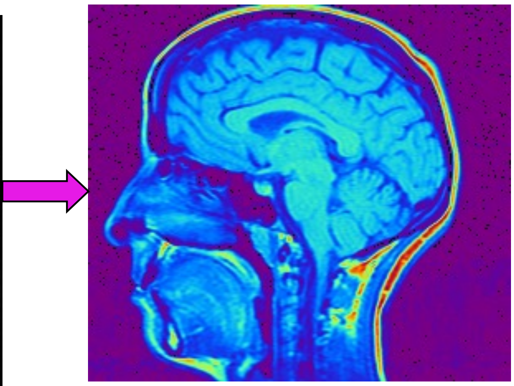


False colors

- Example



Original



False colors

Correlation

2D correlation

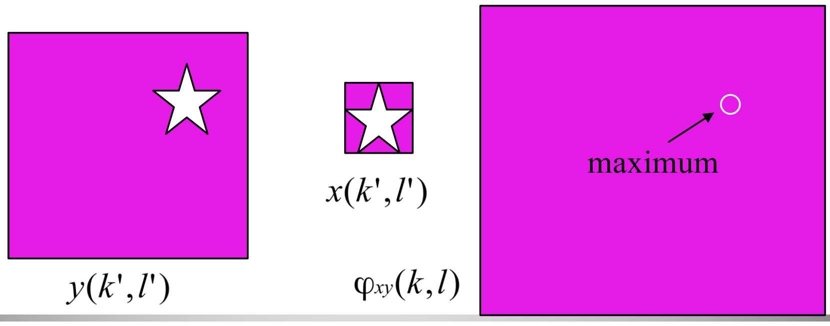
- 2D (inter)-correlation function

$$\varphi_{xy}(k,l) = \sum_{k'=-\infty}^{+\infty} \sum_{l'=-\infty}^{+\infty} x(k',l')\, y(k'+k, l'+l)$$

Correlation

53

- The (inter-) correlation function helps to measure the similarities between two signals
- Application example : Identify if a pattern is present in an image



54

Convolution

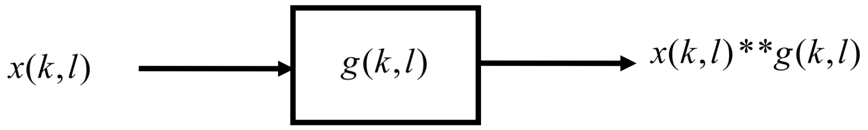
2D convolution

55

- 2D convolution

$$x(k,l)**g(k,l)=\sum_{k'=-\infty}^{+\infty}\sum_{l'=-\infty}^{+\infty}x(k',l')g(k-k',l-l')$$

- Typical representation
  - Signal x is filtered by filter g



Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

55

2D convolution : properties

56

- Commutativity

$$x(k,l)**g(k,l)=\sum_{k'=-\infty}^{+\infty}\sum_{l'=-\infty}^{+\infty}x(k',l')g(k-k',l-l')$$

$$=\sum_{k'=-\infty}^{+\infty}\sum_{l'=-\infty}^{+\infty}g(k',l')x(k-k',l-l')=g(k,l)**x(k,l)$$

Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

56

2D convolution : properties

- Associativity

$$[x(k,l)**g(k,l)]**h(k,l) = x(k,l)**[g(k,l)**h(k,l)]$$

- Distributivity

$$x(k,l)**[g(k,l)+h(k,l)] = x(k,l)**g(k,l) + x(k,l)**h(k,l)$$

Relationship between correlation and convolution

- Correlation can be expressed as a convolution

$$\varphi_{xy}(k,l) = x(-k,-l)**y(k,l)$$

Practical issues on convolution

- In practice, when calculating a convolution product, one has to:
  - Determine the most efficient formula
  - Determine the limits of sums
  - Resolve the border problem

Determine the most efficient formula

- When the size of the image x(k,l) is much more important than the size of the filter g(k,l), the following formula is more appropriate:

$$x(k,l)**g(k,l)=\sum_{k'=-\infty}^{+\infty}\sum_{l'=-\infty}^{+\infty}g(k',l')x(k-k',l-l')$$

Determine the limits of sums

- Limits of sums are directly determined from the size of the filter g(k,l) :

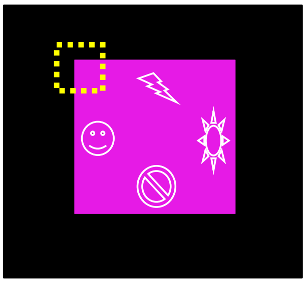
$$x(k,l)**g(k,l) = \sum_{k'=0}^{Mg-1} \sum_{l'=0}^{Ng-1} g(k',l')x(k-k',l-l')$$

Resolve the border problem

- Values of pixels outside of image x(k,l) must be determined when the filter g(k,l) only partially covers the image.
- Several approaches are possible:
  - Zero padding
  - Periodic extension
  - Mirror extension

Zero padding

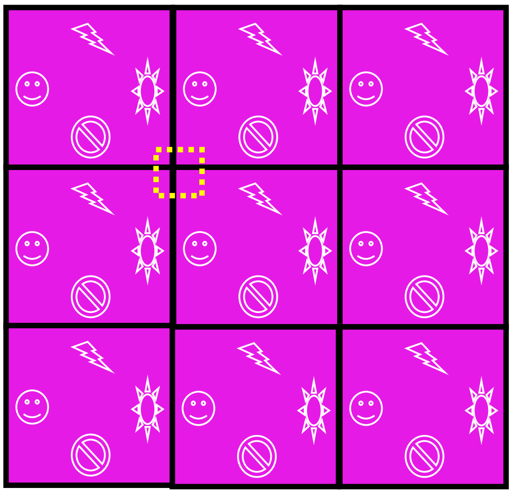
- Simple
- Produces strong border artifacts



Periodic extension

- Simple algorithm
- Coherent with Fourier Transform approach
- Better results when compared to zero padding, if the opposite borders of the image are similar

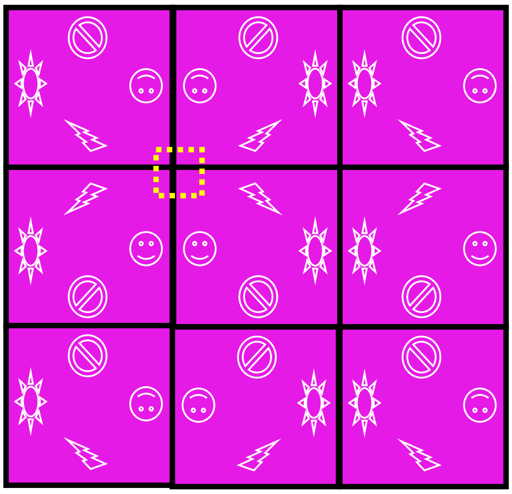
Periodic extension



Mirror extension

- More complex
- Produces the least artifacts

Mirror extension



Z-transform
Fourier Transform

2D Z-transform (2DZT)

- 2DZT of a 2D discrete signal $x(k,l)$

$$X(z_1,z_2)=\sum_{k=-\infty}^{+\infty}\sum_{l=-\infty}^{+\infty}x(k,l)z_1^{-k}z_2^{-l}$$

- Inverse transform

$$x(k,l)=\frac{1}{(2\pi j)^2}\oint_{C_1}\oint_{C_2}X(z_1,z_2)z_1^{k-1}z_2^{l-1}dz_1dz_2$$

2DZT: properties

- Linearity

$$y(k,l)=a_1.x_1(k,l)+a_2.x_2(k,l)$$
$$\Leftrightarrow$$
$$Y(z_1,z_2)=a_1X_1(z_1,z_2)+a_2X_2(z_1,z_2)$$

- Separability

$$x(k,l)=x_1(k).x_2(l)$$
$$\Leftrightarrow$$
$$X(z_1,z_2)=X_1(z_1).X_2(z_2)$$

2DZT: properties

- Translation

$$y(k,l) = x(k-k_0, l-l_0)$$
$$\Leftrightarrow$$
$$Y(z_1, z_2) = z_1^{-k_0} z_2^{-l_0} X(z_1, z_2)$$

- Convolution

$$y(k,l) = x(k,l) ** g(k,l)$$
$$\Leftrightarrow$$
$$Y(z_1, z_2) = X(z_1, z_2).G(z_1, z_2)$$

2DZT: properties

- Conjugate symmetry

$$y(k,l) = x^*(k,l)$$
$$\Leftrightarrow$$
$$Y(z_1, z_2) = X^*(z_1^*, z_2^*)$$

- Symmetry in spatial domain

$$y(k,l) = x(-k,-l)$$
$$\Leftrightarrow$$
$$Y(z_1, z_2) = X(z_1^{-1}, z_2^{-1})$$

2D Continuous Fourier Transform (2DFT)

- 2DFT of a discrete signal $x(k,l)$

$$X(f,g)=\sum_{k=-\infty}^{+\infty}\sum_{l=-\infty}^{+\infty}x(k,l)e^{-j2\pi fk}e^{-j2\pi gl}$$

- Inverse transform

$$x(k,l)=\int_{-1/2}^{1/2}\int_{-1/2}^{1/2}X(f,g)e^{j2\pi fk}e^{j2\pi gl}\,dfdg$$

2DFT: properties

- Periodicity

$$X(f+1,g+1)=X(f,g)$$

- Relationship to 2DZT

$$X(f,g)=X(z_1,z_2)\Big|_{z_1=\exp(j2\pi f),z_2=\exp(j2\pi g)}$$

2DFT: properties

- Symmetrical for all x(k,l) real

$$X(f,g) = X^*(-f,-g)$$

$\mathrm{Re}[X(f,g)]\ ,\ |X(f,g)|$ Even functions

$\mathrm{Im}[X(f,g)]\ ,\ \arg[X(f,g)]$ Odd functions

$x(k,l)$: real and even $\Leftrightarrow X(f,g)$: real and even

$x(k,l)$: real and odd $\Leftrightarrow X(f,g)$: purely imaginary and odd

Some definitions related to 2DFT

- Transfer Function $X(f,g)$

$$X(f,g) = |X(f,g)|e^{j\arg[X(f,g)]}$$

- Amplitude spectrum $|X(f,g)|$
- Phase spectrum $\arg[X(f,g)]$
- Energy (power) spectrum $|X(f,g)|^2$

2DFT: phase et magnitude

- The phase of 2DFT contains information about edges and contours of an image
  - ⇒ Image structure is very much visible in the phase of its 2DFT
- Theoretically, an image can be reconstructed from its phase or its magnitude only
  - – But for magnitude, it's more « difficult »

2DFT: importance of magnitude information

$$x'(k,l) = F^{-1}\left[| F[x(k,l)] | \right]$$

(logarithm of the intensity)

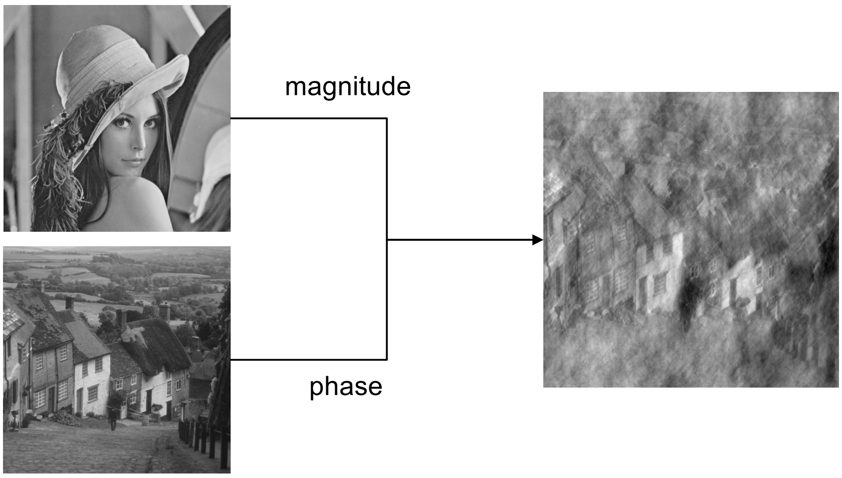
2DFT: importance of phase information

79

$$x'(k,l) = \mathrm{Re}\left[F^{-1}\left[e^{j \arg F[x(k,l)]}\right]\right]$$

2DFT: fusion of magnitude and phase

80



Discrete 2D Fourier Transform: D2DFT

- D2DFT is obtained by sampling in the frequency domain, the 2DFT of a discrete signal, with the following conditions:

$$f = m\Delta f \qquad g = n\Delta g \qquad \text{avec} \quad \Delta f = \frac{1}{K} \quad \text{et} \quad \Delta g = \frac{1}{L}$$

D2DFT: Formula

- Forward transform

$$X(m,n) = \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} x(k,l) \exp[-2j\pi(\frac{mk}{K} + \frac{nl}{L})]$$

- Inverse transform

$$x(k,l) = \frac{1}{KL} \sum_{m=0}^{K-1} \sum_{n=0}^{L-1} X(m,n) \exp[2j\pi(\frac{mk}{K} + \frac{nl}{L})]$$

D2DFT: properties

- All properties of 2DFT remain valid for D2DFT
- The sum of the coefficients of a digital filter provides its frequency response at the origin:

$$X(0,0) = \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} x(k,l)$$

D2DFT: multiplication and circular convolution

- Multiplication of D2DFTs produces a circular convolution !!!

$$x(k,l)\#\#\, y(k,l) \xleftrightarrow{TF} X(m,n)Y(m,n)$$

$$x(k,l)\#\#\, y(k,l) = \sum_{m=0}^{K} \sum_{n=0}^{L} x(m,n)\, y((k-m) \bmod K, (l-n) \bmod L)$$

D2DFT: fast transform and filtering

- A fast implementation of D2DFT is possible if the dimensions of the signal are in powers of 2
- Complexity:

$$O(N^2) \Rightarrow O(N \log N)$$

⇒ This represents an interest in frequency domain filtering

2D digital filtering

2D digital filtering

- In practice three approaches can be used to perform filtering operations
  - Filtering by convolution (direct method)
  - Filtering in the transform domain (Fourier)
  - Filtering by differential equations

Linear filters, transfer functions, frequency responses

- A linear and delay invariant filter can be completely characterized by its impulse response $h(k,l)$

$$y(k,l) = h(k,l) ** x(k,l)$$

- The transfer function $H(z_1,z_2)$ is given by the Z-transform of $h(k,l)$

$$Y(z_1,z_2) = H(z_1,z_2)X(z_1,z_2)$$

- The frequency response is given by:

$$H(z_1 = e^{j2\pi f}, z_2 = e^{j2\pi g})$$

Stability issues in digital 2D filters

- A filter is stable if for any finite amplitude input signal, the output is also of finite amplitude
- Stable filter ⇔ Stable impulse response

$$\sum_{k=-\infty}^{+\infty}\sum_{l=-\infty}^{+\infty} |h(k,l)| < \infty$$

- Stable filter ⇔ the unit hyper-sphere ($|z_1|=1,|z_2|=1$) is contained in RoC of $H(z_1,z_2)$

FIR filters

- $h(k,l)$ is a Finite Impulse Response (FIR) filter if it has a finite number of non-zero samples
  - Always stable
- Easy to conceive and to implement
- Very much used in practice
- Typically with odd samples in each dimension and of limited size
  - 3×3, 9×9, …, max. 19×19 or 21×21

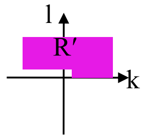
FIR filters of zero phase

- In image processing phase information is very important. It should not be modified by filtering ⇒

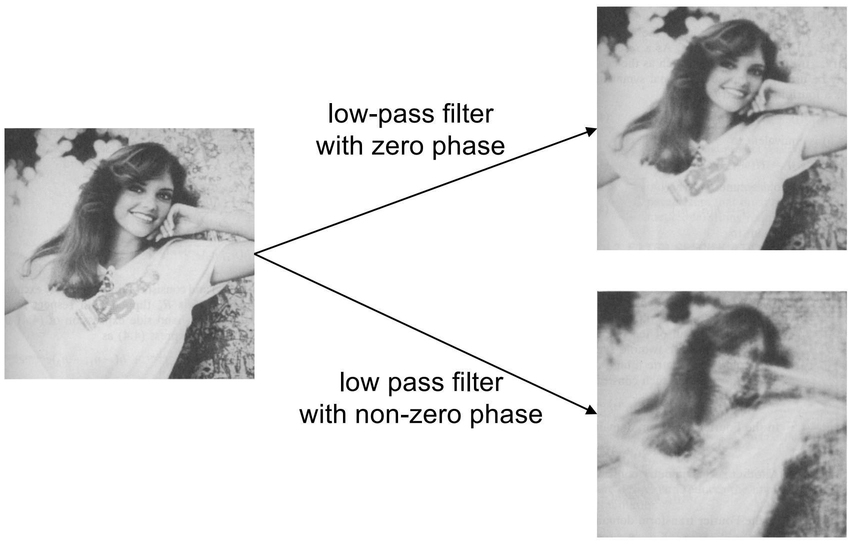
$$h(k,l) = h(-k,-l) \quad \Leftrightarrow \quad H(f,g) = H^*(f,g)$$

- The frequency response:

$$H(f,g) = h(0,0) + \sum_{(k,l)\in R'} 2h(k,l)\cos(2\pi fk + 2\pi gl)$$



FIR filters of zero phase



Filtering by convolution

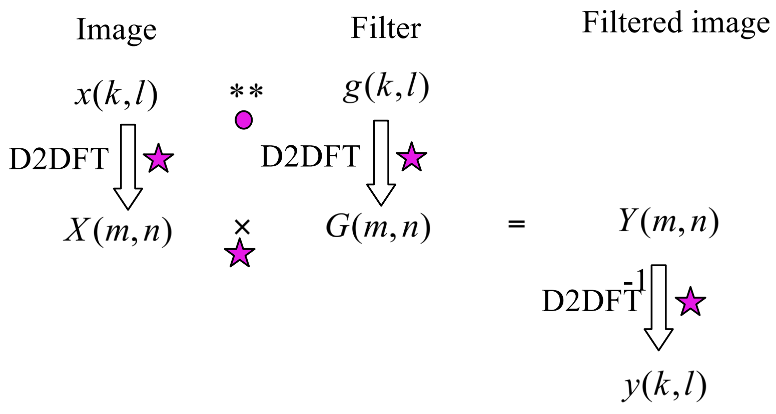
- Easy to implement on DSPs
- Particularly suitable for FIR filters with relatively small dimensions

Frequency domain filtering (Fourier)

- Could be used for both Finite Impulse Response (FIR) and Infinite Impulse Response (IIR)filters
- Advantageous when the size of the image is large and the filter is non-trivial
  - Extension to dimensions in powers of 2

Frequency domain filtering (Fourier)

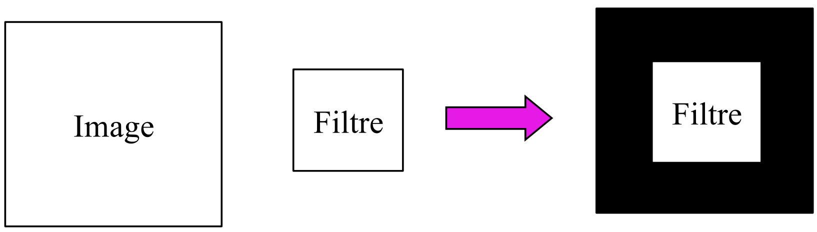
95



Frequency domain filtering (Fourier)

96

- The difference in size between the image and the filter is not a problem
  - the filter is extended to the same size by zero padding



Filering with 2D differential equations

- 2D differential equations are the only way to realize IIR filters

$$\sum\sum_{(m,n)\in R_a} a(m,n)y(k-m,l-n) = \sum\sum_{(m,n)\in R_b} b(m,n)x(k-m,l-n)$$

R_a and R_b are supports of $a(k,l)$ and $b(k,l)$

- Additional boundary conditions are necessary to obtain a unique solution (i.e. a well-defined system)
- Not all boundary conditions result in a linear system

Stability issues in IIR filters

- Transfert function
- $$H(z_1,z_2) = \frac{\sum b(k,l)z_1^{-k}z_2^{-l}}{\sum a(k,l)z_1^{-k}z_2^{-l}} = \frac{B(z_1,z_2)}{A(z_1,z_2)}$$
  - RoC should contain $(|z_1|=1,|z_2|=1)$

FIR versus IIR filters

- FIR filters are always stable
- FIR filters are easier to conceive
- Zero phase FIR filters are more trivial
- IIR filters often require less mathematical operations for a similar frequency response
  - But an implementation in the Fourier domain is often equivalent in terms of complexity
- In practice IIR filters are not often used

Example of filtering - A

$$g(k,l) = \begin{bmatrix} 0 & 1/6 & 0 \\ 1/6 & 1/3 & 1/6 \\ 0 & 1/6 & 0 \end{bmatrix}$$

Example of filtering - A

101

- Frequency response : $G(f,g)$

$$G(f,g)=\sum_{k=-1}^{1}\sum_{l=-1}^{1}g(k,l)e^{-j2\pi fk}e^{-j2\pi gl}$$

$$=\frac{1}{3}+\frac{1}{6}e^{-j2\pi f}+\frac{1}{6}e^{-j2\pi g}+\frac{1}{6}e^{+j2\pi f}+\frac{1}{6}e^{+j2\pi g}$$

$$=\frac{1}{3}+\frac{1}{3}\cos(2\pi f)+\frac{1}{3}\cos(2\pi g)$$

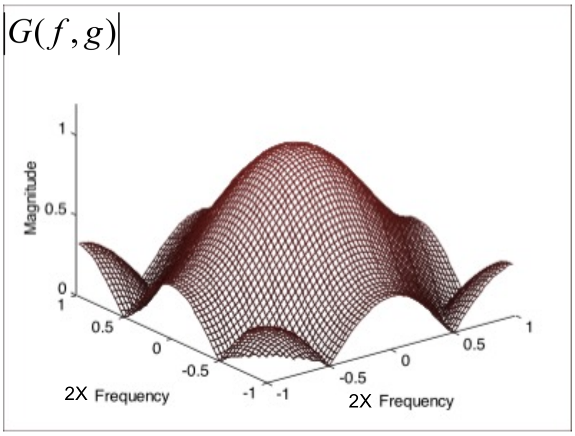
$$G(0,0)=1$$

Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

101

Example of filtering - A

102



Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

102

Example of filtering - A

Implementation by 2-D convolution

$$y(k,l) = 1/6[x(k,l+1) + x(k-1,l)$$
$$+ 2x(k,l) + x(k+1,l) + x(k,l-1)]$$

Extension of the samples in the image beyond its support domain

Example of filtering - B

$$g(k,l) = \begin{bmatrix} 1 & -3 & 1 \\ -3 & 9 & -3 \\ 1 & -3 & 1 \end{bmatrix}$$

Example of filtering - B

- Observation

$$g(k,l) = \begin{bmatrix} 1 & -3 & 1 \\ -3 & 9 & -3 \\ 1 & -3 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 3 \\ -1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -1 \end{bmatrix} = g_1(k).g_1(l)$$

$$g_1(k) = \begin{bmatrix} -1 & 3 & -1 \end{bmatrix}$$

Example of filtering - B

- Frequency response

$$G(f,g) = G_1(f).G_1(g)$$

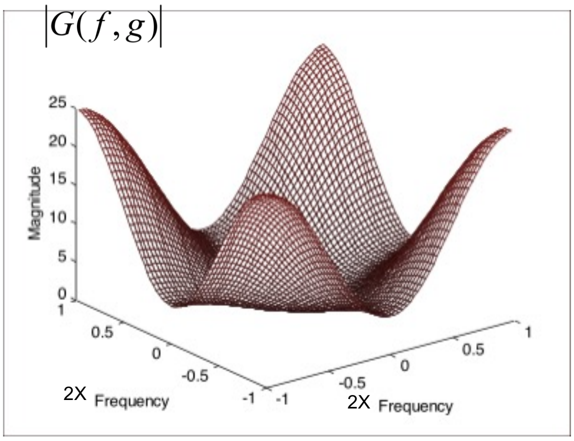
$$G_1(f) = \sum_{k=-1}^{1} g_1(k)e^{-j2\pi fk}$$

$$= -e^{+j2\pi f} + 3 - e^{-j2\pi f} = 3 - 2\cos(2\pi f)$$

$$G(f,g) = [3 - 2\cos(2\pi f)].[3 - 2\cos(2\pi g)]$$

$$G(0,0) = 1$$

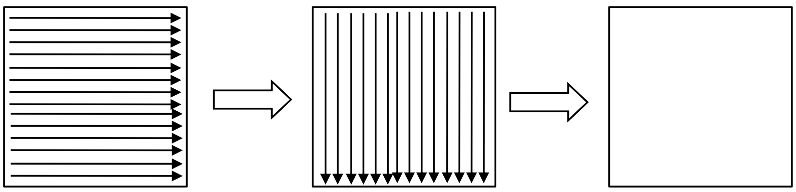
Example of filtering - B



Example of filtering - B

Implementation by 1-D convolution

$$y'(k) = -x'(k-1) + 3x'(k) - x'(k+1)$$



Extension of the samples in the image beyond its support domain

Example of filtering - C

$$g(k,l) = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & \ddots & & & 1 \\ \vdots & & \ddots & & \vdots \\ 1 & & & \ddots & 1 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

15 (rows) × 15 (columns)

Example of filtering - C

- Observation

$$g(k,l) = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & \ddots & & & 1 \\ \vdots & & \ddots & & \vdots \\ 1 & & & \ddots & 1 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \end{bmatrix} = g_1(k).g_1(l)$$

$$g_1(k) = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \end{bmatrix}$$

Example of filtering - C

- Frequency response

$$G(f,g) = G_1(f).G_1(g)$$

$$G_1(f) = \sum_{k=-7}^{7} g_1(k) e^{-j2\pi fk}$$

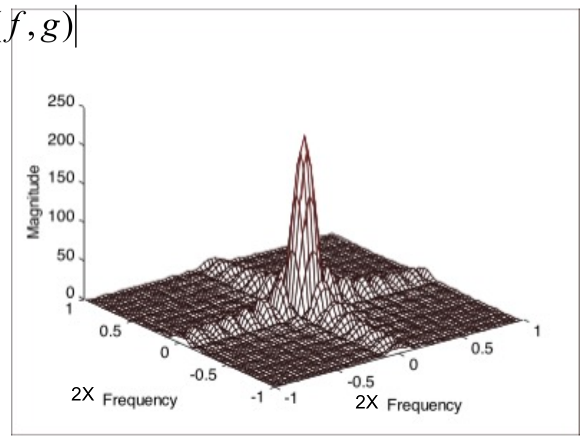
$$= 1 + 2\sum_{n=1}^{7} \cos(2\pi nf)$$

$$G(f,g) = [1 + 2\sum_{n=1}^{7} \cos(2\pi nf)].[1 + 2\sum_{m=1}^{7} \cos(2\pi mg)]$$

$$G(0,0) = 225$$

Example of filtering - C

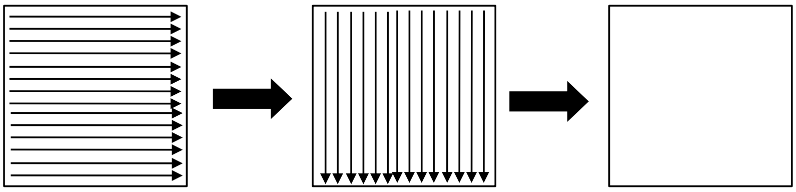
$|G(f,g)|$



Example of filtering - C

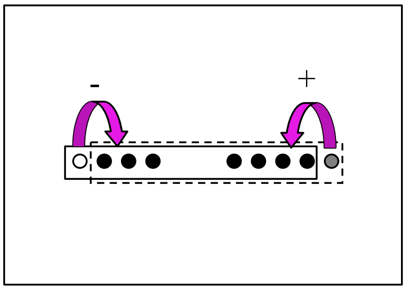
Implementation by 1-D convolution

$$y'(k) = x'(k-7) + \cdots + x'(k) + \cdots + x'(k+7)$$



Extension of the samples in the image beyond its support domain

Example of filtering - C

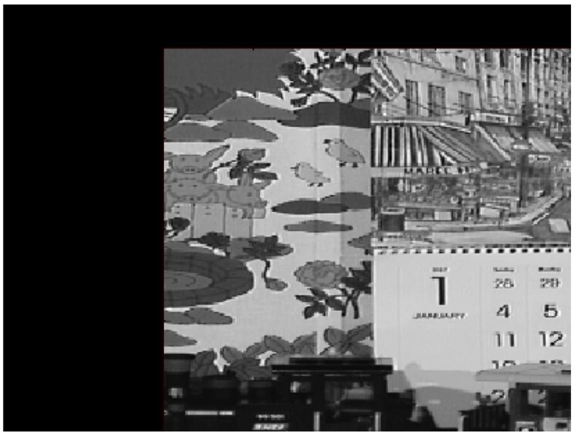


Implementation by 1-D differential equations

$$y'(k+1) = y'(k) + x'(k+8) - x'(k-7)$$

Digital filtering - original

115



Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

115

Digital filtering - filter A

116

Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

116

Digital filtering - filter B

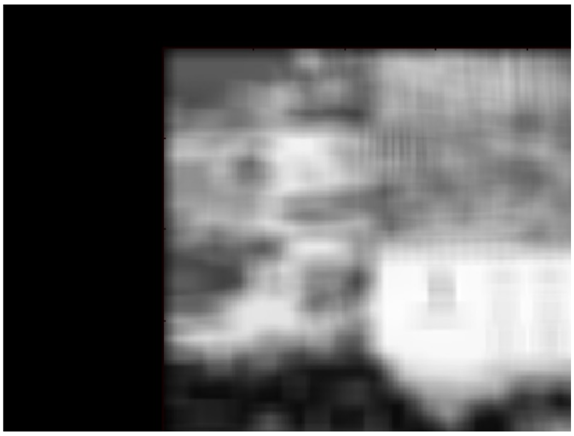
117

Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne

117

Digital filtering - filter C

118



Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne

118

Up-sampling

- Up-sampling of a discrete signal is performed by interleaving a number of zeros between its samples, followed by an ideal low-pass filter
  - Insertion of zero samples produces a compaction and repetition of the spectrum of the input
  - The ideal low-pass filter isolates the main spectrum

Zero order up-sampling

- Replacing zeros interleaving by sample repetition

$$\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \Rightarrow \begin{matrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 3 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 \end{matrix}$$

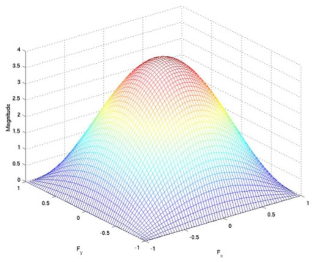
- Similar to use of an “averaging” low-pass filter after zero interleaving

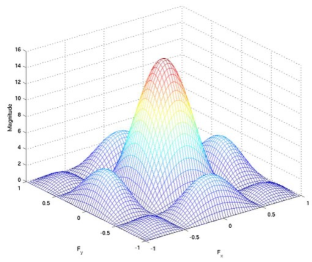
$$\begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}$$

Zero order up-sampling

4x

'averaging' low pass filter





2x2

1	1
1	1

4x4

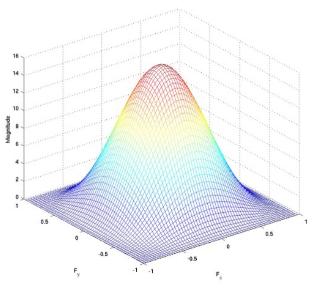
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

➔ introduction of high frequencies!

Polynomial filters of higher orders

- Up-sampling can be improved by use of higher order polynomial interpolation functions
- This is equivalent to application of an "averaging" low-pass filter in a recursive way
- Terminology
  - linear = 'averaging' ** 'averaging'
  - quadratic = linear ** 'averaging'
  - cubic = quadradic ** 'averaging'

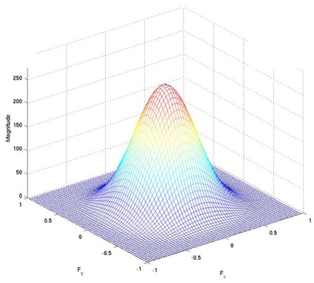
Polynomial filters driven from a 2x2 'averaging'

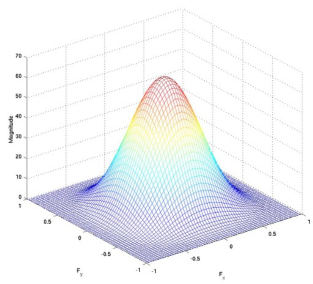


linear

$$\frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 1 & 1 \end{bmatrix}$$

cubic

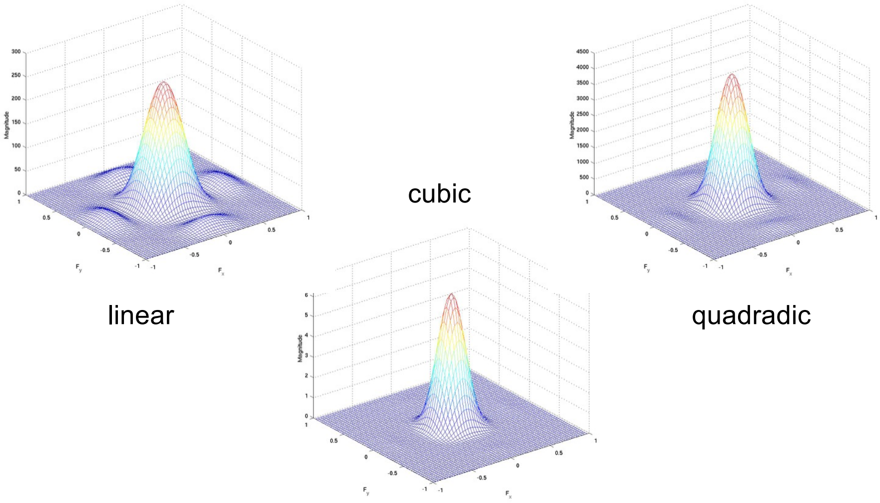




quadradic

Polynomial filters driven from a 4x4 'averaging'

125



Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

125

Gaussian filter

126

- When the order of a polynomial filter increases, it tends to a Gaussian filter

$$G_\sigma(k,l) = \frac{1}{2\sigma^2\pi} e^{-\frac{k^2+l^2}{2\sigma^2}}$$

- Gaussian filter is a separable filter

$$G_\sigma(k,l) = G_\sigma(k) G_\sigma(l) \qquad G_\sigma(k) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{k^2}{2\sigma^2}}$$

Multimedia Signal Processing Group
Ecole Polytechnique Fédérale de Lausanne
EPFL

126

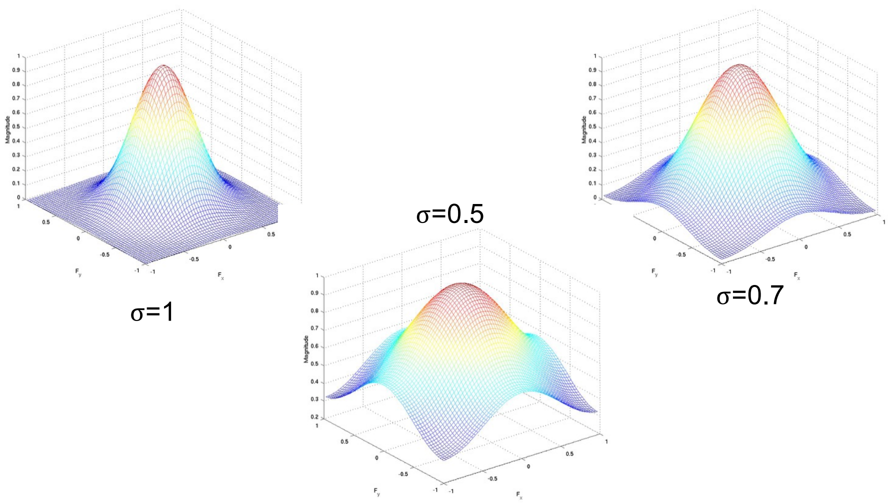
Gaussian filter

127

- In the continuous domain, the Fourier transform of a Gaussian signal is another Gaussian with an inversed standard deviation
  - Infinite support in both space and frequency domains
- In the discrete domain, the Fourier transform of a Gaussian signal is a Gaussian with spectral overlapping

Gaussian filter of size 7x7

128



Gaussian filter: rational approximation

- Coefficients of a Gaussian filter are irrational numbers
  → problem of implementation in integer arithmetics
- A rational approximation is therefore desired
  – This can be achieved by making use of a binomial distribution based on central limit theorem

Gaussian filter: rational approximation

- 1-D case

$$b(s) = \frac{n!}{s!(n-s)!}\frac{1}{2^n}, \qquad s = 0,1,\ldots,n$$

n	normalization factor	coefficients
1	2	1 1
2	4	1 2 1
3	8	1 3 3 1
4	16	1 4 6 4 1
5	32	1 5 10 10 5 1
6	64	1 6 15 20 15 6 1
7	128	1 7 21 35 35 21 7 1

  – 2-D case is obtained by separable extension

Laplacian: approximation

- Laplacian is defined as:

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

- Using a decomposition in Taylor series:

$$f(x+1) \approx f(x) + f'(x) + \tfrac{1}{2} f''(x)$$

one obtains

$$f'(x) \approx f(x) - f(x-1)$$

$$f''(x) \approx 2f(x+1) - 4f(x) + 2f(x-1)$$

Laplacian: approximation

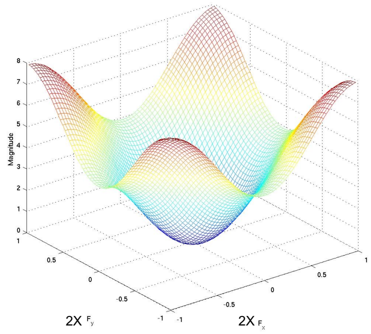
- The following filter is a possible approximation of Laplacian

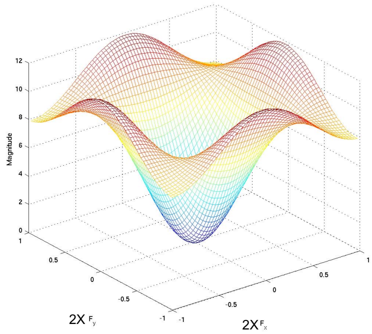
$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

- Other approximations

$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix} \qquad \begin{matrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{matrix} \qquad \frac{4}{\alpha+1}\begin{bmatrix} \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \\ \frac{1-\alpha}{4} & -1 & \frac{1-\alpha}{4} \\ \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \end{bmatrix}$$

Laplacian





$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Laplacian

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix}$$

« Sharpening »

135

- Edges in an image can be enhanced by subtracting from an original signal the result of its Laplacian filtering

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- This would however also enhance noise

« sharpening »

136

Laplacian of a Gaussian

- To avoid problems with noise, Gaussian filtering can be applied before Laplacian

$$\Delta G_\sigma(k,l) = \frac{k^2 + l^2 - 2\sigma^2}{2\pi\sigma^6} e^{-\frac{k^2+l^2}{2\sigma^2}}$$

- The basic principle of "sharpening" remains valid

Sharpening with Laplacian of a Gaussian