Lecture 11: Multimedia Signal Processing

A/Prof. Jian Zhang
NICTA & CSE UNSW
COMP9519 Multimedia Systems
S2 2009
jzhang@cse.unsw.edu.au

11.1 Linear Image Filters and Convolution

- Linear Filtering Process:
  - Form a new image whose pixels are based on some function of a local neighborhood of the pixels
    - Replace each pixel with a linear combination of its neighbors through a function
    - This function is called kernel function which gives the prescription for the linear combination
    - The process of this operation is called convolution

10	5	3
4	5	1
1	1	7

$*$

0.5	0	0.25
0	0.5	0
0.25	0	0.5

kernel

$=$

	12	

11.1 Linear Image Filters and Convolution

- Convolution
  - Centre original of the kernel *F* at each pixel location
  - Multiply weights by the corresponding pixels
  - Set resulting value for each pixel

$$R_{ij} = \sum_{u,v} H_{i-u,j-v} F_{uv}$$

  - Image, *R*, resulting from convolution of *F* with image *H*, Where u,v range over kernel pixels
- For 2D image, the convolution equation is as follows

$$f[m,n] = I \otimes g = \sum_{k,l} I[m-k, n-l] g[k,l]$$

11.1 Linear Image Filters and Convolution

- Mask with positive entries, that sum to 1.
- Replaces each pixel with an average of its neighborhood.
- If all weights are equal, it is called a **box** filter.

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Slide credit: David Lowe (UBC)

11.1 Linear Image Filters and Convolution

Definition: Correlation

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X+i, Y+j)$$

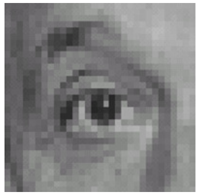
Definition: Convolution

$$I'(X,Y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j)\, I(X-i, Y-j)$$

$$= \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(-i,-j)\, I(X+i, Y+j)$$

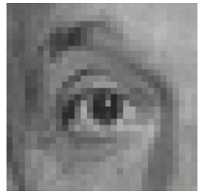
NOTE: If $F(X,Y) = F(-X,-Y)$ then correlation $\equiv$ convolution

11.1 Linear Image Filters and Convolution



Original

0	0	0
0	1	0
0	0	0

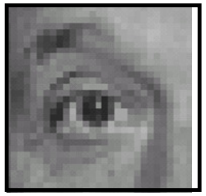


Filtered
(no change)

11.1 Linear Image Filters and Convolution

Original

0	0	0
0	0	1
0	0	0



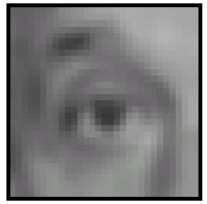
Shifted left
By 1 pixel

11.1 Linear Image Filters and Convolution

Original

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1



Blur (with a
box filter)

11.1 Linear Image Filters and Convolution

Original

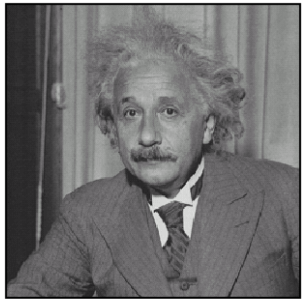
$$\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right)$$

Sharpening filter

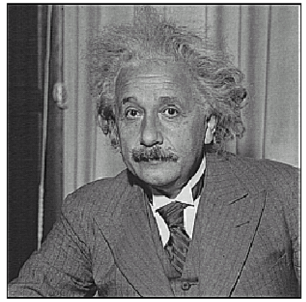
- Accentuates differences with local average
- Also known as Laplacian

11.1 Linear Image Filters and Convolution

Sharpening



before

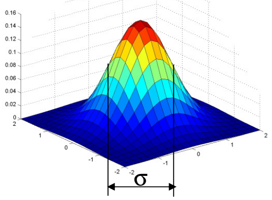


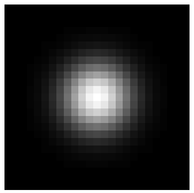
after

11.1 Linear Image Filters and Convolution

- Gaussian Kernel
  - Idea: Weight contributions of neighboring pixels by nearness

Slide credit: Christopher Rasmussen





0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

- Constant factor at front makes volume sum to 1.

11.1 Linear Image Filters and Convolution

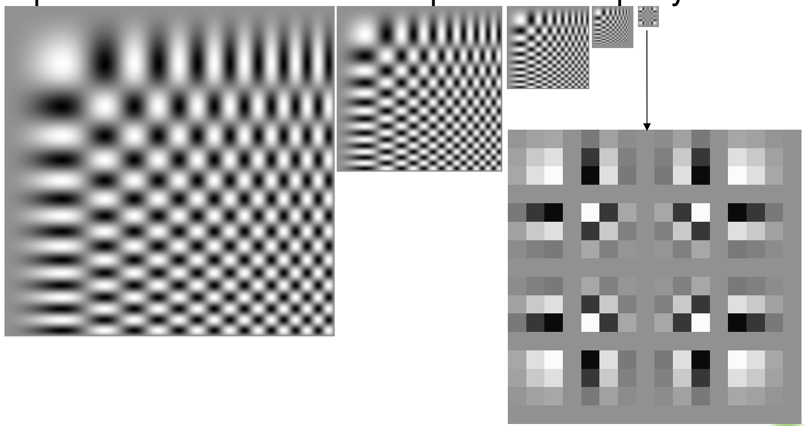
- Gaussian Vs Average

Gaussian Smoothing

Smoothing by Averaging

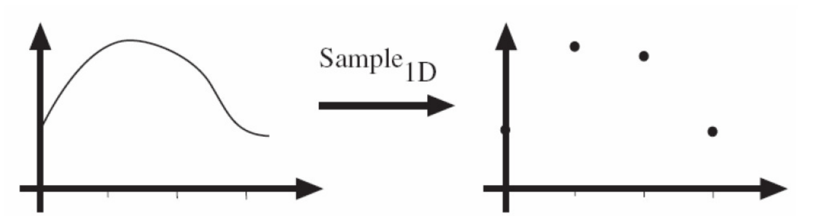
11.2 Image Sampling

- Constructing a pyramid by simply taking every second pixel results in a bad misrepresent at top layer



11.2 Image Sampling

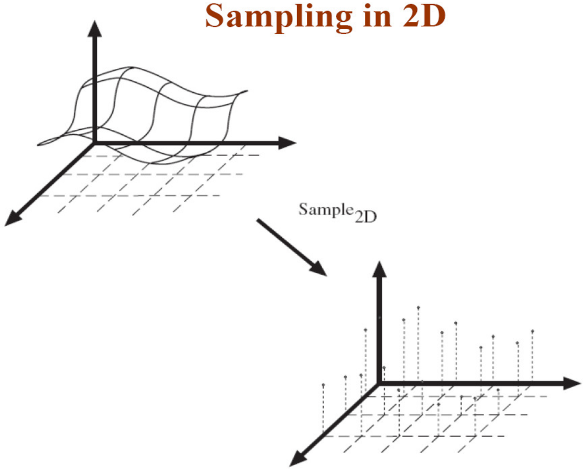
Sampling in 1D



Sampling in 1D takes a function, and returns a vector whose elements are values of that function at the sample points, as the top figures show. For our purpose, it is enough that the sample points be integer values of the argument. We allow the vector to be infinities dimensional, and have negative as well as positive indices

11.2 Image Sampling

Sampling in 2D



11.2 Image Sampling

- A continuous model of a sampled signal

$$\begin{aligned}
\int_{-\infty}^{\infty} a\delta(x)f(x)dx &= a\lim_{\epsilon\to 0}\int_{-\infty}^{\infty} d(x;\epsilon)f(x)dx \\
&= a\lim_{\epsilon\to 0}\int_{-\infty}^{\infty} \frac{bar(x;\epsilon)}{\epsilon}(f(x))dx \\
&= a\lim_{\epsilon\to 0}\sum_{i=-\infty}^{\infty} \frac{bar(x;\epsilon)}{\epsilon}(f(i\epsilon)bar(x-i\epsilon;\epsilon))\epsilon \\
&= af(0)
\end{aligned}$$

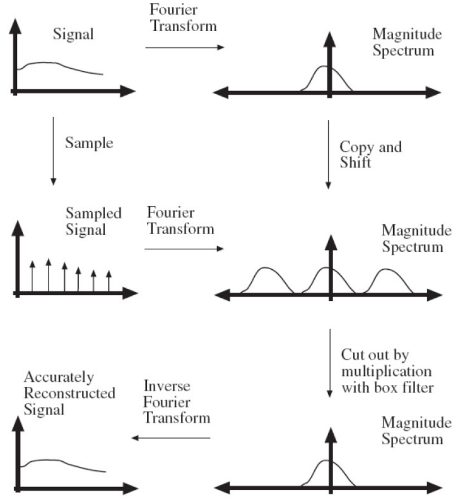
$$\begin{aligned}
\mathtt{sample}_{2D}(f) &= \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} f(x,y)\delta(x-i,y-j) \\
&= f(x,y)\left\{\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(x-i,y-j)\right\}
\end{aligned}$$

11.2 Image Sampling

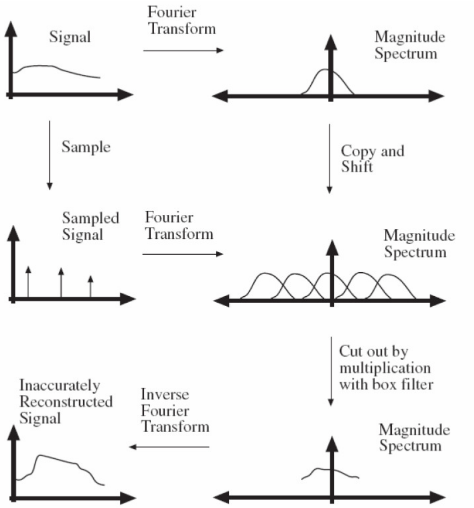
- Fourier Transform (FT) of a Sampled Signal

$$\mathcal{F}(\texttt{sample}_{2D}(f(x,y))) = \mathcal{F}\left(f(x,y)\left\{\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(x-i,y-j)\right\}\right)$$
$$= \mathcal{F}(f(x,y)) * \mathcal{F}\left(\left\{\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\delta(x-i,y-j)\right\}\right)$$
$$= \sum_{\substack{i=-\infty \\ j=-\infty}}^{\infty} F(u-i, v-j)$$

11.2 Image Sampling



11.2 Image Sampling



11.2 Image Sampling

- Nyquist's theorem
  - In order for a band-limited (i.e., one with a zero power spectrum for frequencies $f > B$) baseband ($f > 0$) signal to be reconstructed fully, it must be sampled at a rate $f \geq 2B$.
  - A signal sampled at $f = 2B$ is known as Nyquist sampled and $f = 2B$ is the Nyquist (NT) frequency.
  - No information is lost if a signal is sampled at the Nyquist frequency, and no additional information is gained by sampling faster than this rate.

11.2 Image Sampling

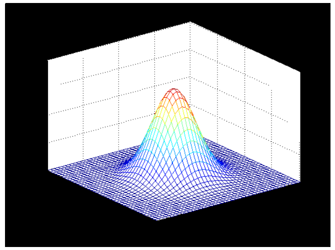
- Smoothing as low-pass filtering
  What NT means that the high frequencies in the signal waves lead to trouble with sampling.
  - Solution: suppress high frequencies before sampling
    - multiply the FT of the signal with something that suppresses high frequencies
    - or convolve with a low-pass filter
  - Common solution: use a Gaussian
    - multiplying FT by Gaussian is equivalent to convolving image with Gaussian.

11.2 Image Sampling
Gaussian Filter

Convolving image (I) with Gaussian

$$G \oplus I$$

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right) \oplus 2D\,\mathrm{Image}$$



$$H * I$$

$$H(i,j) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{((i-k-1)^2+(j-k-1)^2)}{2\sigma^2}\right)$$

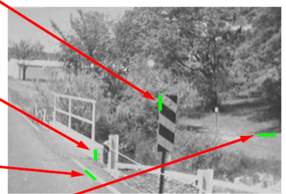
where $H(i,j)$ is $(2k+1)\times(2k+1)$ array

11.3 Edge detection

- **Goal:**
  - Identify sudden changes (discontinuities) in an image
- This is where most shape information is encoded

What causes an edge?

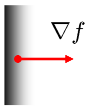
- Depth discontinuity
- Surface orientation discontinuity
- Reflectance discontinuity (i.e., change in surface material properties)
- Illumination discontinuity (e.g., shadow)



Slide credit: Christopher Rasmussen

11.3 Edge detection
Image gradient

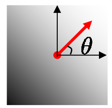
- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$
- The gradient points in the direction of most rapid change in intensity



$\nabla f = \left[\frac{\partial f}{\partial x}, 0\right]$

2.3 Edge detection

$\nabla f = \left[0, \frac{\partial f}{\partial y}\right]$



$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$

- The gradient direction is given by:

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

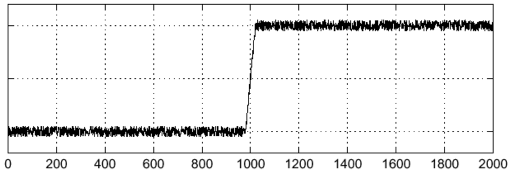
  - how does this relate to the direction of the edge?
- The *edge strength* is given by the gradient magnitude

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

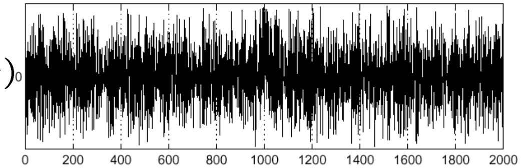
11.3 Edge detection
Effects of noise

- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal

$f(x)$



$\frac{d}{dx}f(x)$



- Where is the edge?

NICTA

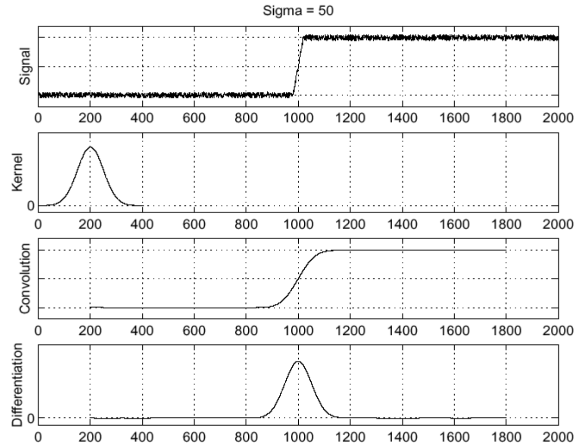
11.3 Edge detection
Solution: smooth first

f

h

$h \star f$

$\frac{\partial}{\partial x}(h \star f)$



- Where is the edge?
- Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

NICTA

11.3 Edge detection
Derivative theorem of convolution

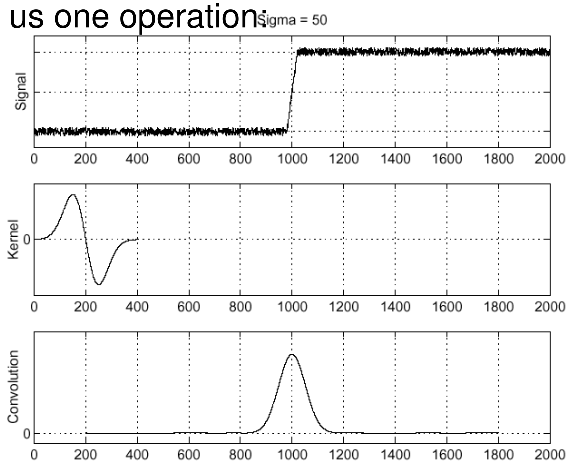
$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

- This saves us one operation:

f

$\frac{\partial}{\partial x}h$

$(\frac{\partial}{\partial x}h) \star f$



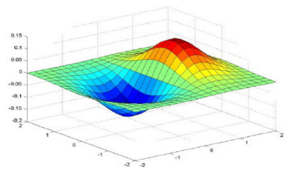
NICTA

11.3 Edge detection
Smoothing and Differentiation

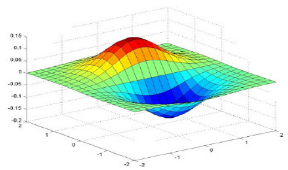
- **Edge:** a location with high gradient (derivative)
- Need smoothing to reduce noise prior to taking derivative
- Need two derivatives, in x and y direction.
- We can use derivative of Gaussian filters
  - because differentiation is convolution, and convolution is associative:

D * (G * I) = (D * G) * I

Ref : Christopher Rasmussen



$\frac{\partial}{\partial x}G_\sigma$



$\frac{\partial}{\partial y}G_\sigma$

NICTA

11.3 Edge detection

original

11.3 Edge detection

smoothed
(5x5 Gaussian)

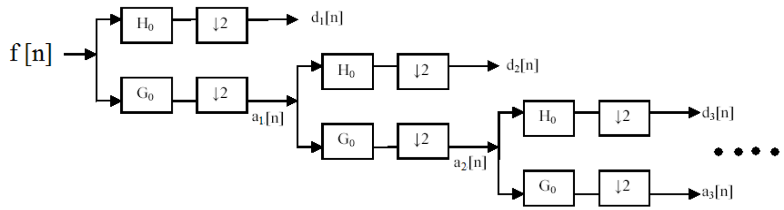
11.3 Edge detection
Subtraction

smoothed – original
(scaled by 4, offset +128)

Why does this work?

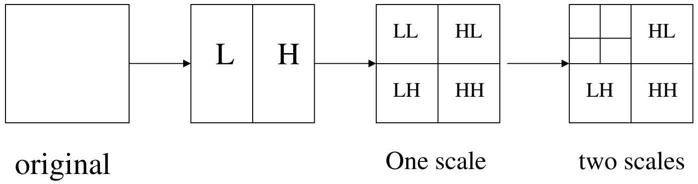
11.4 2-D Transform
1D Discrete Wavelet Transform (DWT)

- Separates the high and low-frequency portions of a signal through the use of filters
- One level of transform:
  - Signal is passed through G & H filters.
  - Down sample by a factor of two
- Multiple levels (scales) are made by repeating the filtering and decimation process on lowpass outputs

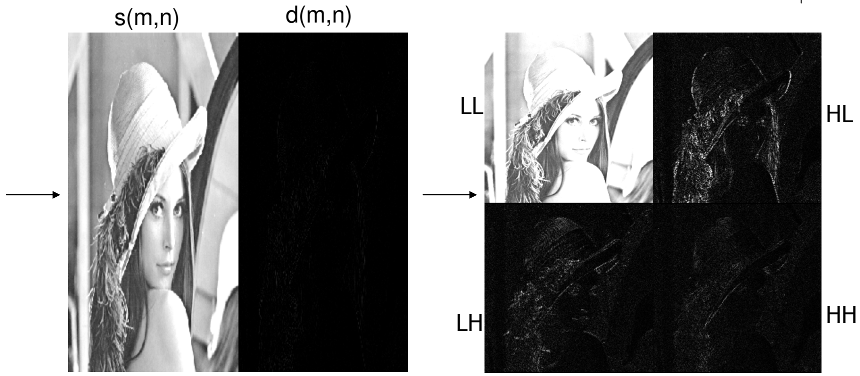


11.4 2D Transform
2-D DWT

- Step 1: replace each row with its 1-D DWT.
- Step 2: Replace each column with its 1-D DWT
- Step 3: Repeat steps 1 & 2 on the lowest subband for the next scale.
- Step 4: Repeat step 3 until as many scales as desired



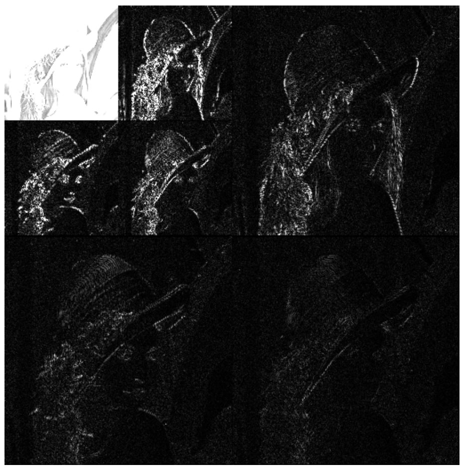
11.4 2D Transform
From 1D to 2D



After row transform:
each row is decomposed into low-band (approximation) and high-band (detail)

Note that the order of row/column transform does not matter

11.4 2D Transform
From 1-level to Multi-level



11.4 2D Transform
Reconstruction

- How those components can be assembled back into the original signal without loss of information?
- A Process After *decomposition* or *analysis*
- Also called *synthesis*
- Reconstruct the signal from the wavelet coefficients
- Where wavelet analysis involves filtering and down-sampling, the wavelet reconstruction process consists of up-sampling and filtering