CS559 – Lecture 5
Sampling, Reconstruction, Resampling

These are course notes (not used as slides)
Written by Mike Gleicher, Sept. 2005
With some slides adapted from the notes of Stephen Chenney

Last time

- Sampling
  - Bad sampling is Bad
  - How to sample correctly
  - Some signal processing theory
- Today: Applying signal processing for sampling
  - Convolution: how we implement signal processing
  - Filtering
  - Reconstruction
  - Resampling

Sampling Theorem

- If your signal is bandlimited
- And you know what the band limit is
- And you sample at (at least) twice that frequency
  - Above the Nyquist rate
- Then – you can reconstruct your signal EXACTLY!
- Caveat
  - Ideal reconstruction requires perfect band limiting in both sampling and reconstruction

Need to know about convolutions

- We need to have band limited signals
  - Need low pass filters
  - Which are implemented as **convolutions**
- Reconstruction requires low-pass filtering
  - Which is implemented as **convolution**
- Need to see Sampling theory in Fourier domain
  - Need **convolution**

Filtering: Convolutions

- A general filter is a function on an image that produces another image
- Many common filters are simpler in the Fourier domain
- Choice:
  - Transform image, filter, inverse transform image
  - Inverse transform operator, apply in spatial domain
  - Transform (or inverse) of multiplication is convolution

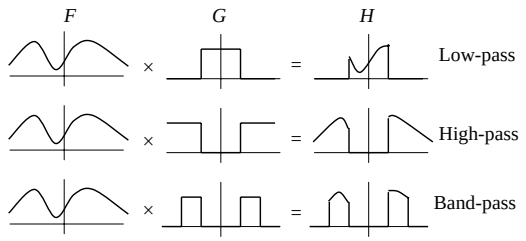
Filters

- A *filter* is something that attenuates or enhances particular frequencies
- Easiest to visualize in the frequency domain, where filtering is defined as multiplication:

$$H(\omega) = F(\omega) \times G(\omega)$$

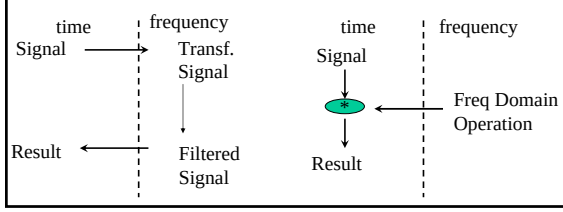
- Here, *F* is the spectrum of the function, *G* is the spectrum of the filter, and *H* is the filtered function. Multiplication is point-wise

Qualitative Filters



Can you transform an operator?

- Many filters are multiplication in frequency domain
- Fourier transform of multiplication is convolution
- Fourier transform of convolution is multiplication



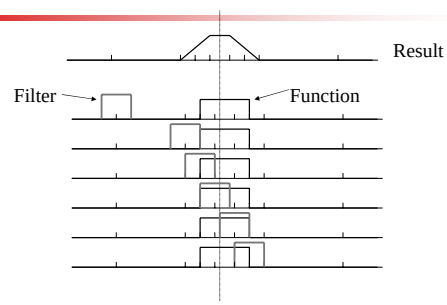
Filtering in the Spatial Domain

- Filtering the spatial domain is achieved by *convolution*

$$h(x) = f \otimes g = \int_{-\infty}^{\infty} f(u)g(x-u)du$$

- Qualitatively: Slide the filter to each position, *x*, then sum up the function multiplied by the filter at that position

Convolution Example



Convolution Theorem

- Convolution in the spatial domain is the same as multiplication in the frequency domain
  - Take a function, *f*, and compute its Fourier transform, *F*
  - Take a filter, *g*, and compute its Fourier transform, *G*
  - Compute $H=F \times G$
  - Take the inverse Fourier transform of *H*, to get *h*
  - Then $h=f \otimes g$
- Multiplication in the spatial domain is the same as convolution in the frequency domain

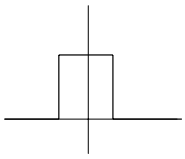
Filtering Images

- Work in the discrete spatial domain
- Convert the filter into a matrix, the *filter mask*
- Move the matrix over each point in the image, multiply the entries by the pixels below, then sum
  - eg 3x3 box filter
  - Effect is averaging

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

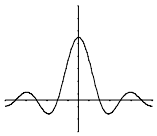
Box Filter

- Box filters smooth by averaging neighbors
- In frequency domain, keeps low frequencies and attenuates (reduces) high frequencies, so clearly a low-pass filter



Spatial: Box

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Frequency: sinc

Filter Widths

- Fourier Transform of a Time scaling:
  - f(k t) -> F(1/k omega)
  - As time gets scaled, frequency gets scaled by the inverse
- Box filter: wider box in frequency domain = narrower filter in time domain
- To filter higher frequencies use a narrow (in time/space) filter
- Lower Frequency cutoff (in a High-pass filter), you use a bigger (in time/space) filter

Handling Boundaries

$$I_{output}[x][y] = \sum_{i=-k/2}^{k/2} \sum_{j=-k/2}^{k/2} I_{input}[x+i][y+j] M[i+k/2][j+k/2]$$

- At (0,0) for instance, you might need pixel data for (-1,-1), which doesn't exist
- Option 1: Make the output image smaller – don't evaluate pixels you don't have all the input for
- Option 2: Replicate the edge pixels
  - Equivalent to: posn = x + i; if (posn < 0) posn = 0; and so on for other indices
- Option 3: Reflect image about edge
  - Equivalent to: posn = x + i; if (posn < 0) posn = -posn; and similar for others

Seperable Filters

- Some 2D filters can be implemented as 2 1D filters
- Each dimension at a time
- Much easier
  - Don't need to build 2D filter kernel
  - Much faster (O(mn) not O(m^2 n))
- Box filters are seperable
- Other 2D filters are designed by seperated pieces

Constructing Masks: 2D

- Multiply 2 1D masks together using *outer product*

$$M[i][j] = m[i]m[j]$$

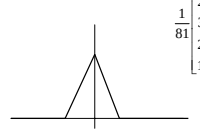
- *M* is 2D mask, *m* is 1D mask

	0.2	0.6	0.2
0.2	0.04	0.12	0.04
0.6	0.12	0.36	0.12
0.2	0.04	0.12	0.04

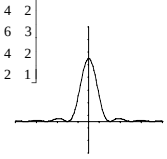
Bartlett Filter

- Triangle shaped filter in spatial domain
- In frequency domain, product of two box filters, so attenuates high frequencies more than a box

$$\frac{1}{81}\begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$$



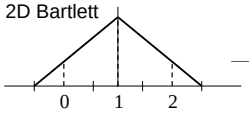
Spatial: Triangle (Box⊗Box)



Frequency: sinc^2

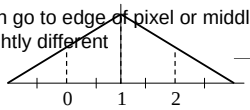
Constructing Masks: 1D

- Sample the filter function at matrix “pixels”, then normalize
- eg 2D Bartlett



$$\frac{1}{5}\begin{bmatrix}1 & 3 & 1\end{bmatrix}$$

- Can go to edge of pixel or middle of next: results are slightly different

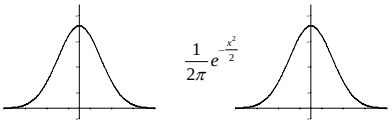


$$\frac{1}{4}\begin{bmatrix}1 & 2 & 1\end{bmatrix}$$

Gaussian Filter

$$\frac{1}{256}\begin{bmatrix}1 & 4 & 6 & 4 & 1\\ 4 & 16 & 24 & 16 & 4\\ 6 & 24 & 36 & 24 & 6\\ 4 & 16 & 24 & 16 & 4\\ 1 & 4 & 6 & 4 & 1\end{bmatrix}$$

- Attenuates high frequencies even further
- In 2d, rotationally symmetric, so fewer artifacts



$$\frac{1}{2\pi}e^{-\frac{x^2}{2}}$$

Constructing Gaussian Mask

- Use the binomial coefficients
  - Central Limit Theorem (probability) says that with more samples, binomial converges to Gaussian

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

$$\frac{1}{16}\begin{bmatrix}1 & 4 & 6 & 4 & 1\end{bmatrix}$$

$$\frac{1}{64}\begin{bmatrix}1 & 6 & 15 & 20 & 15 & 6 & 1\end{bmatrix}$$

```
  1 1
 1 2 1
1 3 3 1
1 4 6 4 1
```

Sampling Theory

- Sampling is multiply by spike chain in time domain
  - Fourier transform of spike chain is spike chain
  - Fourier transform of multiply is convolution
- Sampling is convolution by spike chain in frequency
- Makes infinite copies of signal
- Reconstruction low-pass filters to remove all but one
- Non-band limited, things “spill”