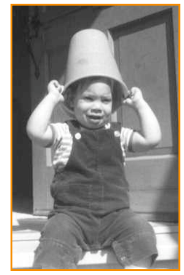
Image Processing

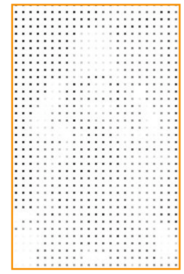
Adam Finkelstein & Tim Weyrich
Princeton University
COS 426, Spring 2008

What is an Image?

- An image is a discrete array of samples representing a continuous 2D function

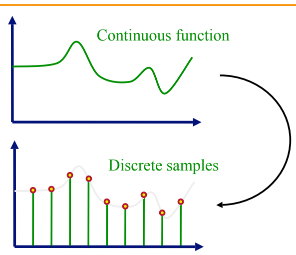


Continuous function

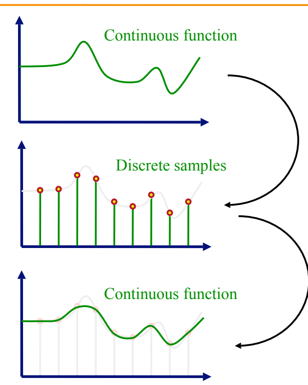


Discrete samples

Sampling and Reconstruction



Sampling and Reconstruction



Sampling and Reconstruction

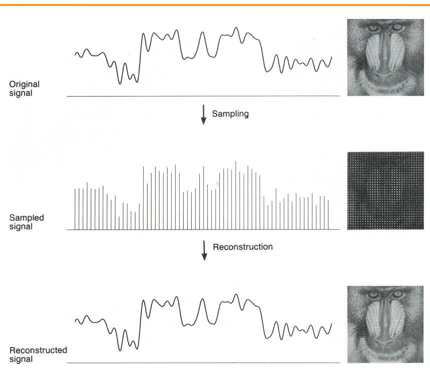
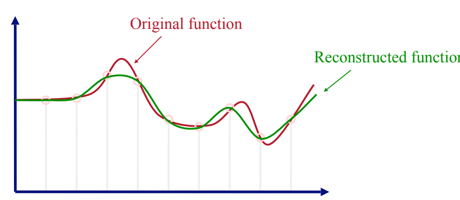


Figure 19.9 FvDFH

Sampling Theory

- How many samples are enough?
  - o How many samples are required to represent a given signal without loss of information?
  - o What signals can be reconstructed without loss for a given sampling rate?



Sampling Theory

- What happens when use too few samples?
  - o Aliasing

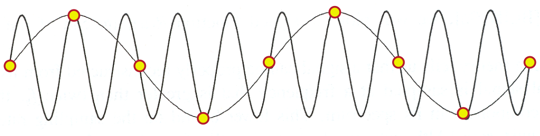
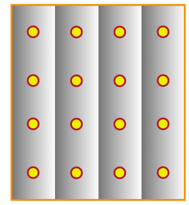
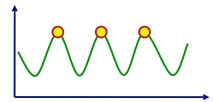


Figure 14.17 FvDFH

Sampling Theory

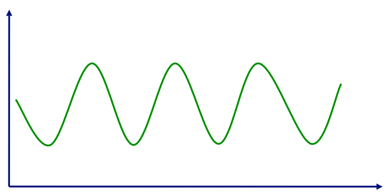
- What happens when use too few samples?
  - o Aliasing





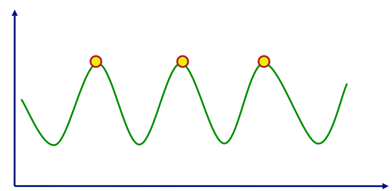
Sampling Theory

- How many samples are enough to avoid aliasing?
  - o How many samples are required to represent a given signal without loss of information?
  - o What signals can be reconstructed without loss for a given sampling rate?



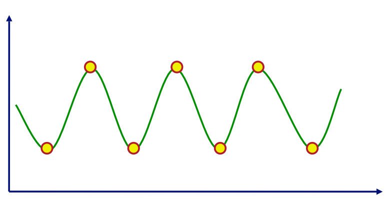
Sampling Theory

- How many samples are enough to avoid aliasing?
  - o How many samples are required to represent a given signal without loss of information?
  - o What signals can be reconstructed without loss for a given sampling rate?



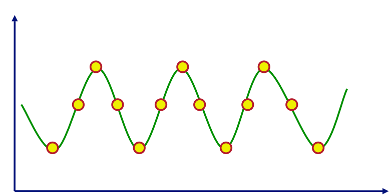
Sampling Theory

- How many samples are enough to avoid aliasing?
  - o How many samples are required to represent a given signal without loss of information?
  - o What signals can be reconstructed without loss for a given sampling rate?



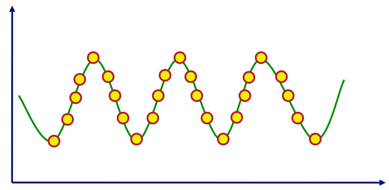
Sampling Theory

- How many samples are enough to avoid aliasing?
  - o How many samples are required to represent a given signal without loss of information?
  - o What signals can be reconstructed without loss for a given sampling rate?



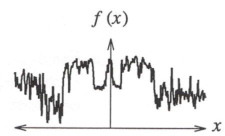
Sampling Theory

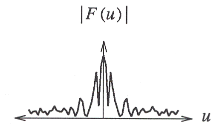
- How many samples are enough to avoid aliasing?
  - How many samples are required to represent a given signal without loss of information?
  - What signals can be reconstructed without loss for a given sampling rate?



Spectral Analysis

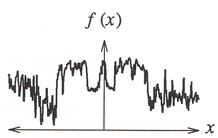
- Spatial domain:
  - Function: f(x)
  - Filtering: convolution
- Frequency domain:
  - Function: F(u)
  - Filtering: multiplication

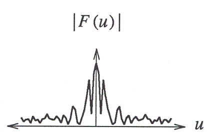




Any signal can be written as a sum of periodic functions.

Fourier Transform





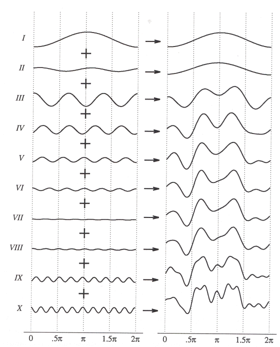


Figure 2.6 Wolberg

Fourier Transform

- Fourier transform:

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi xu} dx$$

- Inverse Fourier transform:

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{+i2\pi ux} du$$

Sampling Theorem

- A signal can be reconstructed from its samples, if the original signal has no frequencies above 1/2 the sampling frequency - Shannon
- The minimum sampling rate for bandlimited function is called “Nyquist rate”

A signal is bandlimited if its highest frequency is bounded. The frequency is called the bandwidth.

Image Processing

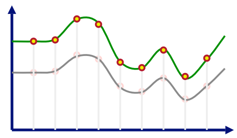
- Pixel operations
  - Add random noise
  - Add luminance
  - Add contrast
  - Add saturation
- Filtering
  - Blur
  - Detect edges
  - Sharpen
  - Emboss
  - Median
- Quantization
  - Uniform Quantization
  - Floyd-Steinberg dither
- Warping
  - Scale
  - Rotate
  - Warp
- Combining
  - Composite
  - Morph

Adjusting Brightness

- Simply scale pixel components
  - o Must clamp to range (e.g., 0 to 1)


Original


Brighter



Adjusting Contrast

- Compute mean luminance L for all pixels
  - o luminance = 0.30*r + 0.59*g + 0.11*b
- Scale deviation from L for each pixel component
  - o Must clamp to range (e.g., 0 to 1)


Original


More Contrast

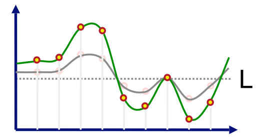
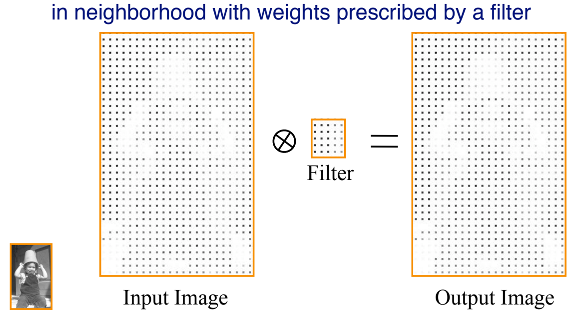


Image Processing

- Pixel operations
  - o Add random noise
  - o Add luminance
  - o Add contrast
  - o Add saturation
- Filtering
  - o Blur
  - o Detect edges
  - o Sharpen
  - o Emboss
  - o Median
- Quantization
  - o Uniform Quantization
  - o Floyd-Steinberg dither
- Warping
  - o Scale
  - o Rotate
  - o Warp
- Combining
  - o Composite
  - o Morph

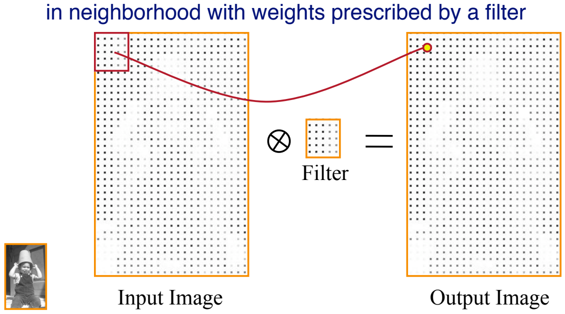
Linear Filtering

- Convolution
  - o Each output pixel is a linear combination of input pixels in neighborhood with weights prescribed by a filter



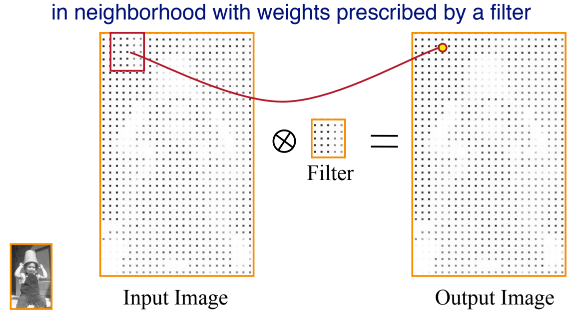
Linear Filtering

- Convolution
  - o Each output pixel is a linear combination of input pixels in neighborhood with weights prescribed by a filter



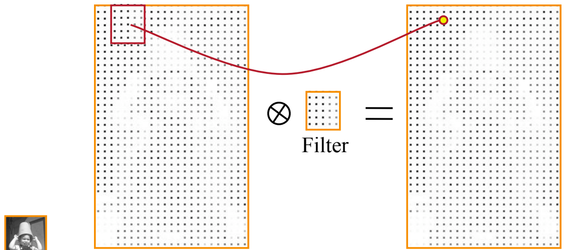
Linear Filtering

- Convolution
  - o Each output pixel is a linear combination of input pixels in neighborhood with weights prescribed by a filter



Linear Filtering

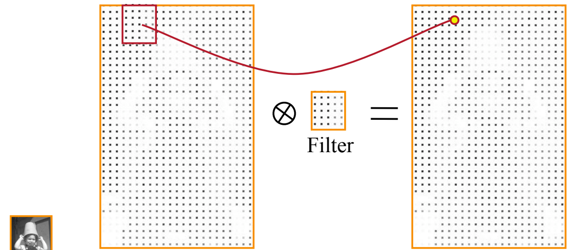
- Convolution
  - o Each output pixel is a linear combination of input pixels in neighborhood with weights prescribed by a filter



Input Image   Output Image

Linear Filtering

- Convolution
  - o Each output pixel is a linear combination of input pixels in neighborhood with weights prescribed by a filter



Input Image   Output Image

Adjust Blurriness

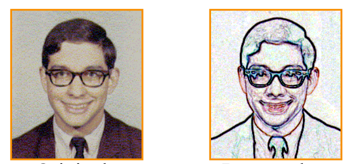
- Convolve with a filter whose entries sum to one
  - o Each pixel becomes a weighted average of its neighbors


Original   Blur

$$\text{Filter} = \begin{bmatrix} 1/16 & 2/16 & 1/16 \\ 2/16 & 4/16 & 2/16 \\ 1/16 & 2/16 & 1/16 \end{bmatrix}$$

Edge Detection

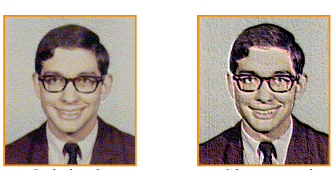
- Convolve with a filter that finds differences between neighbor pixels


Original   Detect edges

$$\text{Filter} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & +8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Sharpen

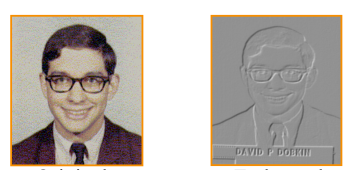
- Sum detected edges with original image


Original   Sharpened

$$\text{Filter} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & +9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Emboss

- Convolve with a filter that highlights gradients in particular directions


Original   Embossed

$$\text{Filter} = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Non-Linear Filtering

- Each output pixel is a non-linear function of input pixels in neighborhood (filter depends on input)


Original


Oil

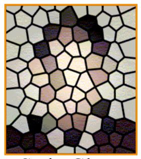
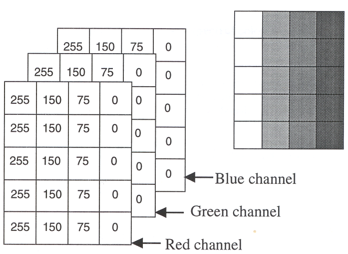

Stain Glass

Image Processing

- Pixel operations
  - o Add random noise
  - o Add luminance
  - o Add contrast
  - o Add saturation
- Filtering
  - o Blur
  - o Detect edges
  - o Sharpen
  - o Emboss
  - o Median
- Quantization
  - o Uniform Quantization
  - o Floyd-Steinberg dither
- Warping
  - o Scale
  - o Rotate
  - o Warp
- Combining
  - o Composite
  - o Morph

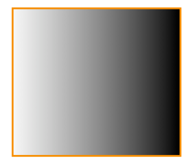
Quantization

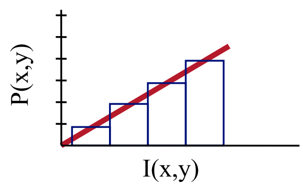
- Reduce intensity resolution
  - o Frame buffers have limited number of bits per pixel
  - o Physical devices have limited dynamic range

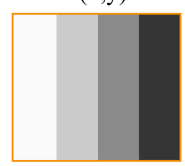


Uniform Quantization

P(x, y) = round(I(x, y))
where round() chooses nearest value that can be represented.


I(x,y)

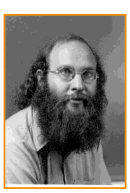



P(x,y)
(2 bits per pixel)

Uniform Quantization

- Images with decreasing bits per pixel:


8 bits


4 bits


2 bits


1 bit

Notice contouring.

Reducing Effects of Quantization

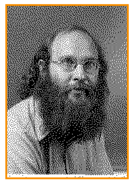
- Dithering
  - o Random dither
  - o Ordered dither
  - o Error diffusion dither
- Halftoning
  - o Classical halftoning

Dithering

- Distribute errors among pixels
  - o Exploit spatial integration in our eye
  - o Display greater range of perceptible intensities

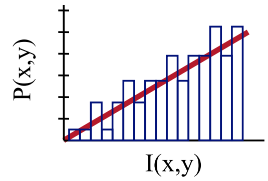

Original
(8 bits)

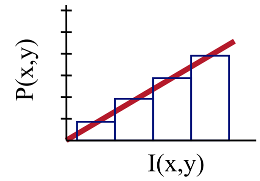

Uniform
Quantization
(1 bit)


Floyd-Steinberg
Dither
(1 bit)

Random Dither

- Randomize quantization errors
  - o Errors appear as noise





P(x, y) = round(I(x, y) + noise(x,y))

Random Dither


Original
(8 bits)


Uniform
Quantization
(1 bit)


Random
Dither
(1 bit)

Ordered Dither

- Pseudo-random quantization errors
  - o Matrix stores pattern of threshholds

i = x mod n
j = y mod n
e = I(x,y) - trunc(I(x,y))
threshold = $(D(i,j)+1)/(n^2+1)$
if (e > threshold)
    P(x,y) = ceil(I(x, y))
else
    P(x,y) = floor(I(x,y))

$$D_2 = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}$$

0 1/5 2/5 3/5 4/5 1
thresholds

Ordered Dither

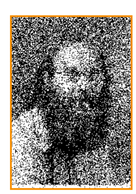
- Bayer's ordered dither matrices

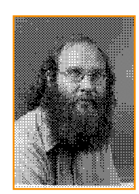
$$D_n = \begin{bmatrix} 4D_{n/2} + D_2(1,1)U_{n/2} & 4D_{n/2} + D_2(1,2)U_{n/2} \\ 4D_{n/2} + D_2(2,1)U_{n/2} & 4D_{n/2} + D_2(2,2)U_{n/2} \end{bmatrix}$$

$$D_2 = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \qquad D_4 = \left[\begin{array}{cc|cc} 15 & 7 & 13 & 5 \\ 3 & 11 & 1 & 9 \\ \hline 12 & 4 & 14 & 6 \\ 0 & 8 & 2 & 10 \end{array}\right]$$

Ordered Dither

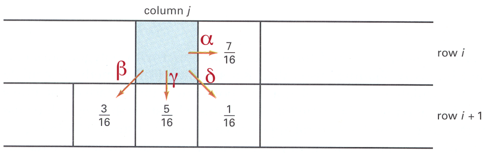

Original
(8 bits)


Random
Dither
(1 bit)


Ordered
Dither
(1 bit)

Error Diffusion Dither

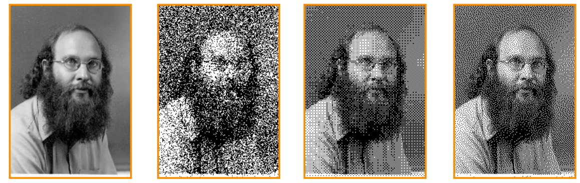
- Spread quantization error over neighbor pixels
  - o Error dispersed to pixels right and below



$$\alpha + \beta + \gamma + \delta = 1.0$$

Figure 14.42 from H&B

Error Diffusion Dither



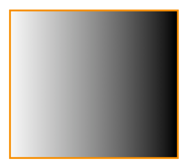
Original (8 bits) | Random Dither (1 bit) | Ordered Dither (1 bit) | Floyd-Steinberg Dither (1 bit)

Reducing Effects of Quantization

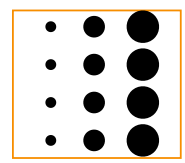
- Dithering
  - o Random dither
  - o Ordered dither
  - o Error diffusion dither
- ➢ Halftoning
  - o Classical halftoning

Classical Halftoning

- Use dots of varying size to represent intensities
  - o Area of dots proportional to intensity in image

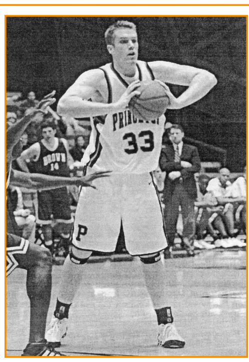


I(x,y)

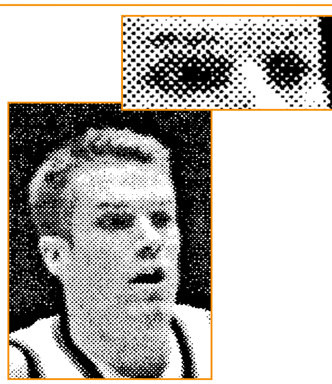


P(x,y)

Classical Halftoning

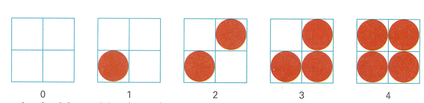


From *Town Topics*, Princeton



Halftone patterns

- Use cluster of pixels to represent intensity
  - o Trade spatial resolution for intensity resolution



Q: In this case, would we use four "halftoned" pixels in place of one original pixel?

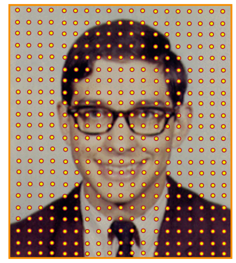
Figure 14.37 from H&B

Image Processing

- Pixel operations
  - o Add random noise
  - o Add luminance
  - o Add contrast
  - o Add saturation
- Filtering
  - o Blur
  - o Detect edges
  - o Sharpen
  - o Emboss
  - o Median
- Quantization
  - o Uniform Quantization
  - o Floyd-Steinberg dither
- Warping
  - o Scale
  - o Rotate
  - o Warp
- Combining
  - o Composite
  - o Morph

Image Processing

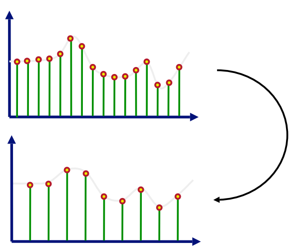
- Consider reducing the image resolution


Original image


1/4 resolution

Image Processing

- Image processing is a resampling problem

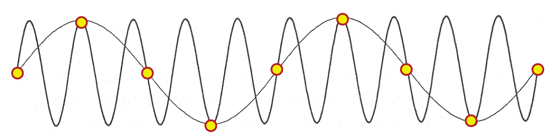


Resampling

Sampling Theorem

- A signal can be reconstructed from its samples, if the original signal has no frequencies above 1/2 the sampling frequency - Shannon

Aliasing will occur if the signal is under-sampled

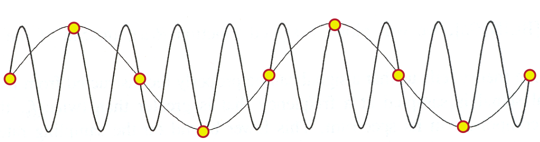


Under-sampling

Figure 14.17 FvDFH

Aliasing

- In general:
  - o Artifacts due to under-sampling or poor reconstruction
- Specifically, in graphics:
  - o Spatial aliasing
  - o Temporal aliasing

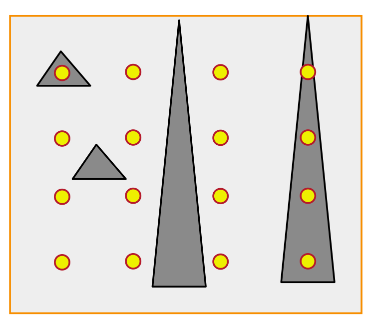


Under-sampling

Figure 14.17 FvDFH

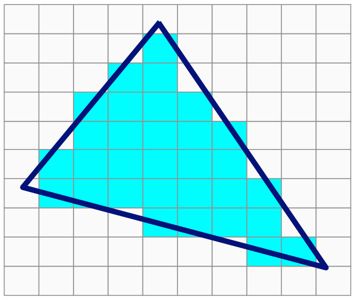
Spatial Aliasing

- Artifacts due to limited spatial resolution



Spatial Aliasing

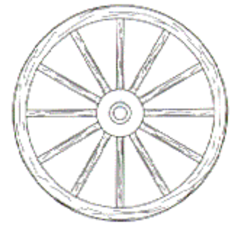
- Artifacts due to limited spatial resolution



"Jaggies"

Temporal Aliasing

- Artifacts due to limited temporal resolution
  - o Strobing
  - o Flickering

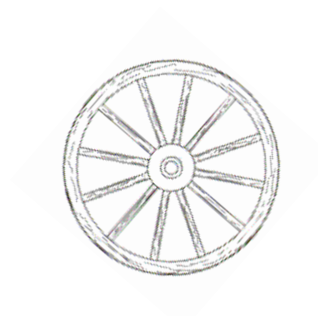


Temporal Aliasing

- Artifacts due to limited temporal resolution
  - o Strobing
  - o Flickering

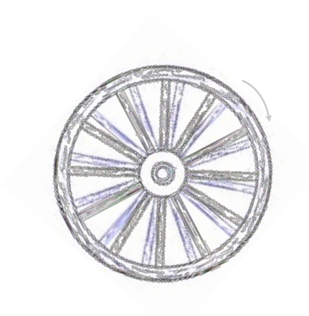
Temporal Aliasing

- Artifacts due to limited temporal resolution
  - o Strobing
  - o Flickering



Temporal Aliasing

- Artifacts due to limited temporal resolution
  - o Strobing
  - o Flickering



Antialiasing

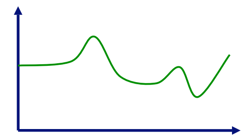
- Sample at higher rate
  - o Not always possible
  - o Doesn't always solve problem
- Pre-filter to form bandlimited signal
  - o Form bandlimited function using low-pass filter
  - o Trades aliasing for blurring

Image Processing

Real world
Sample
Discrete samples (pixels)
Reconstruct
Reconstructed function
Transform
Transformed function
Filter
Bandlimited function
Sample
Discrete samples (pixels)
Reconstruct
Display

Image Processing

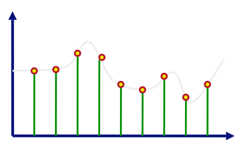
Real world
Sample
Discrete samples (pixels)
Reconstruct
Reconstructed function
Transform
Transformed function
Filter
Bandlimited function
Sample
Discrete samples (pixels)
Reconstruct
Display



Continuous Function

Image Processing

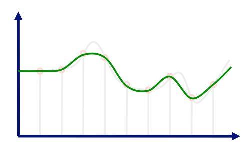
Real world
Sample
Discrete samples (pixels)
Reconstruct
Reconstructed function
Transform
Transformed function
Filter
Bandlimited function
Sample
Discrete samples (pixels)
Reconstruct
Display



Discrete Samples

Image Processing

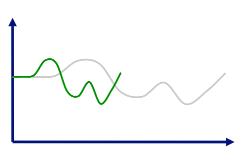
Real world
Sample
Discrete samples (pixels)
Reconstruct
Reconstructed function
Transform
Transformed function
Filter
Bandlimited function
Sample
Discrete samples (pixels)
Reconstruct
Display



Reconstructed Function

Image Processing

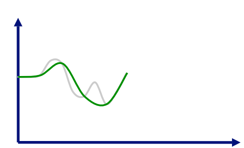
Real world
Sample
Discrete samples (pixels)
Reconstruct
Reconstructed function
Transform
Transformed function
Filter
Bandlimited function
Sample
Discrete samples (pixels)
Reconstruct
Display



Transformed Function

Image Processing

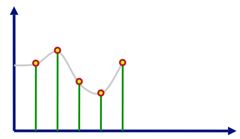
Real world
Sample
Discrete samples (pixels)
Reconstruct
Reconstructed function
Transform
Transformed function
Filter
Bandlimited function
Sample
Discrete samples (pixels)
Reconstruct
Display



Bandlimited Function

Image Processing

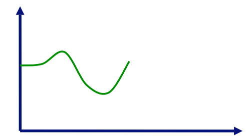
Real world
Sample
Discrete samples (pixels)
Reconstruct
Reconstructed function
Transform
Transformed function
Filter
Bandlimited function
Sample
Discrete samples (pixels)
Reconstruct
Display



Discrete samples

Image Processing

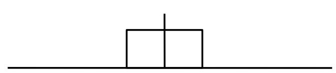
Real world
Sample
Discrete samples (pixels)
Reconstruct
Reconstructed function
Transform
Transformed function
Filter
Bandlimited function
Sample
Discrete samples (pixels)
Reconstruct
Display



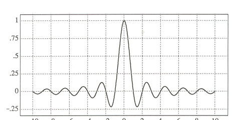
Display

Ideal Bandlimiting Filter

- Frequency domain



- Spatial domain



$$Sinc(x) = \frac{\sin \pi x}{\pi x}$$

Figure 4.5 Wolberg

Practical Image Processing

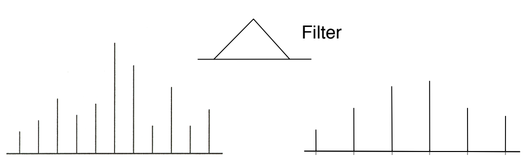
- Finite low-pass filters
  - o Point sampling (bad)
  - o Triangle filter
  - o Gaussian filter

Real world
Sample
Discrete samples (pixels)
Reconstruct
Reconstructed function
Transform
Transformed function
Filter
Bandlimited function
Sample
Discrete samples (pixels)
Reconstruct
Display

Convolution

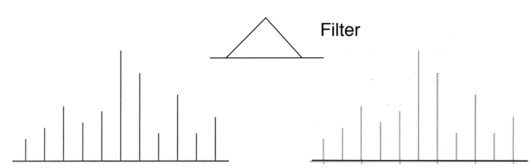
Convolution

- Spatial domain: output pixel is weighted sum of pixels in neighborhood of input image
  - o Pattern of weights is the “filter”



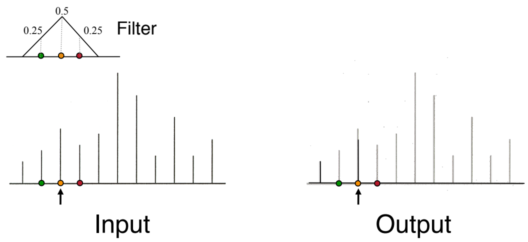
Convolution with a Triangle Filter

- Example 1:



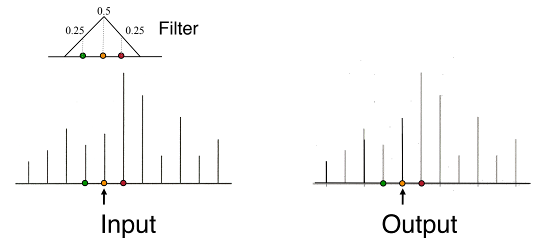
Convolution with a Triangle Filter

- Example 1:



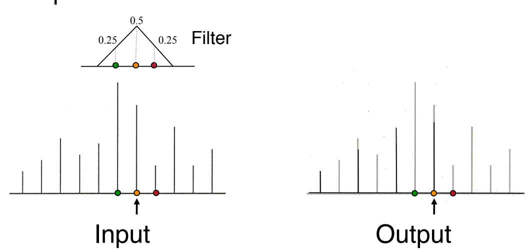
Convolution with a Triangle Filter

- Example 1:



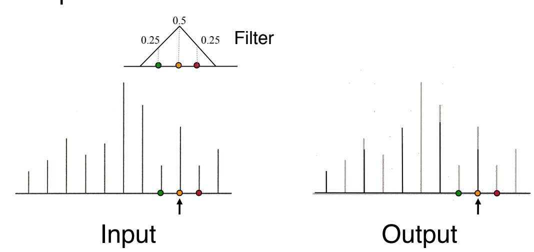
Convolution with a Triangle Filter

- Example 1:



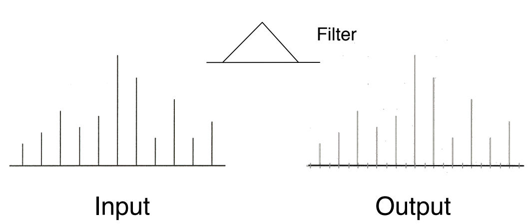
Convolution with a Triangle Filter

- Example 1:



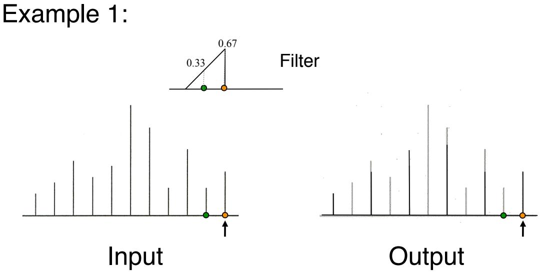
Convolution with a Triangle Filter

- Q: What if the filter runs off the end?



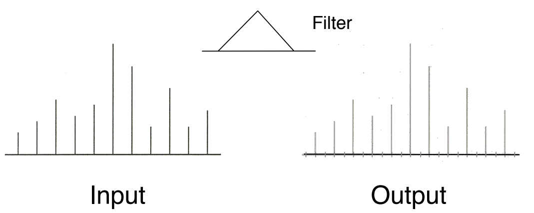
Convolution with a Triangle Filter

- Example 1:



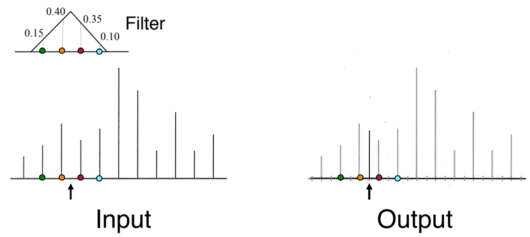
Convolution with a Triangle Filter

- Q: what if the filter is not centered on a sample?



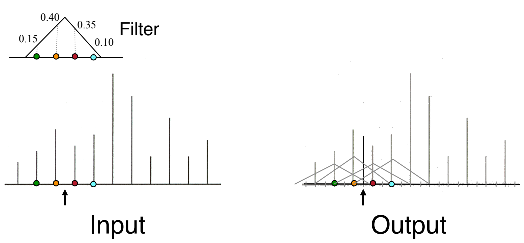
Convolution with a Triangle Filter

- Example 2:



Convolution with a Triangle Filter

- Example 2:



Convolution with a Triangle Filter

- Example 3 (triangle filter of radius 1):

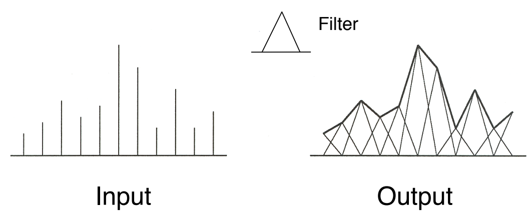


Figure 2.4 Wolberg

Convolution with a Gaussian Filter

- Example 4:

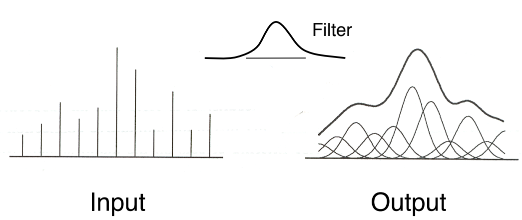


Figure 2.4 Wolberg

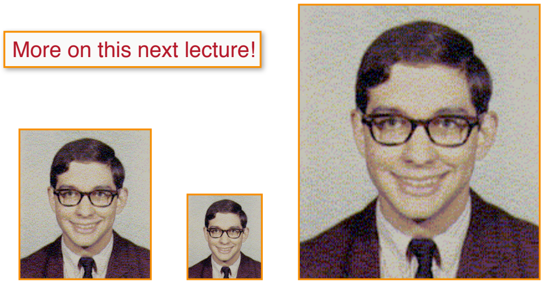
Image Processing

- Pixel operations
  - o Add random noise
  - o Add luminance
  - o Add contrast
  - o Add saturation
- Filtering
  - o Blur
  - o Detect edges
  - o Sharpen
  - o Emboss
  - o Median
- Quantization
  - o Uniform Quantization
  - o Floyd-Steinberg dither
- Warping
  - o Scale
  - o Rotate
  - o Warp
- Combining
  - o Composite
  - o Morph

Scaling

- Resample with triangle or Gaussian filter

More on this next lecture!



Original | 1/4X resolution | 4X resolution

Summary

- Image filtering
  - o Compute new values for image pixels based on function of old values
- Halftoning and dithering
  - o Reduce visual artifacts due to quantization
  - o Distribute errors among pixels
    - » Exploit spatial integration in our eye
- Sampling and reconstruction
  - o Reduce visual artifacts due to aliasing
  - o Filter to avoid undersampling
    - » Blurring is better than aliasing