Last time

- We learned how to draw
  - Point
  - Line
  - Circle
- Avoid floating operations for speed
Today

- How to fill a 2D polygon?
- Image processing
Image Processing & Antialiasing
Jaggies & Aliasing

- Why Jaggies?
  - Represent continuous geometry by discrete 2D pixels

- Anti-aliasing smoothes pixels around the jaggies
  - Shades of gray instead of sharp black/white transitions
Midpoint algorithm
In each column, pick the closest to the line

A form of point sampling:
sample the line at each of the integer X values

Doubling resolution in x and y only lessens the problem, but costs 4 times memory, bandwidth, and scan conversion time!
Representing lines: Area sampling

- Represent the line as a unit width rectangle, use multiple pixels overlapping the rectangle (for now we think of pixels as squares)

- Instead of full on/off, calculate each pixel intensity proportional to the area covered by the unit rectangle

- A form of unweighted area sampling:
  - Only pixels covered by primitive can contribute
  - Distance of pixel center to line doesn’t matter

- Typically have more than one pixel per column
“Box Filter” Represents Unweighted Area Sampling

- Box filter is constant over all area and is a single pixel wide here, but could vary in width.
- For each pixel intersecting the line, intensity contributed by each sub-area of intersection \( dA \) is \( W(x, y) dA \).
- Then total intensity of the pixel (between 0 and 1) integrated over area of overlap is:
  \[
  \int_A W(x, y) dA
  \]
- Integral is volume over area of overlap (in above figure, a rectangular wedge).
“Cone Filter” for Weighted Area Sampling

- Area sampling, but the overlap between filter and primitive is weighted so that sub-areas with $dA$ closer to center of pixel count more

- Cone has:
  - Linear falloff
  - Circular symmetry
  - Width of 2 (called **support**)

- Intensity of pixel is the “subvolume” inside the cone over the line (see picture)
Another Look at Unweighted Area Sampling: Box filter

- **Box filter**
  - Support: 1 pixel
  - Sets intensity proportional to area of overlap
  - Creates “winking” of adjacent pixels as a small triangle translates

Unweighted area sampling. (a) All sub-areas in the pixel are weighted equally. (b) Changes in computed intensities as an object moves between pixels.
Another Look at Weighted Area Sampling: Pyramid filter

- Pyramid filter
  - Support: 1 pixel
  - Approximates circular cone to emphasize area of overlap close to center of pixel

Weighted area sampling. (a) sub-areas in the pixel are weighted differently as a function of distance to the center of the pixel. (b) Changes in computed intensities as an object moves between pixels.
Another Look at Weighted Area Sampling: Cone filter

- Cone filter
  - Support: 2 pixels
  - Greater smoothness in the changes of intensity

Weighted area sampling with overlap. (a) Typical weighting function. (b) Changes in computed intensities as an object moves between pixels.
Another Look at Point Sampling – Even Box Filter is Better!

Point-sampling problems. Samples are shown as black dots. Object $A$ and $C$ are sampled, but corresponding objects $B$ and $D$ are not.

- This simplistic scan conversion algorithm only asks if a mathematical point is inside the primitive or not.
This simplistic scan conversion algorithm only asks if a mathematical point is inside the primitive or not.
This simplistic scan conversion algorithm only asks if a mathematical point is inside the primitive or not.
Point sampling vs Area sampling

for each sample point $p$: // $p$ need not be integer!

place filter centered over $p$

for each pixel $q$ under filter:

weight = filter value over $q$

$p$.intensity += weight * $q$.intensity

Aliased

Anti-aliased
Moire Pattern
Fourier Transform and Nyquist Limit
Sampling of Images

- Scan converting an image is digitizing (sampling) a series of continuous intensity functions, one per scan line.
- We will use single scan lines for simplicity, but everything still applies to images.

Scan line from synthetic scene

Scan line from natural scene
The Sampling/Reconstruction/Display Pipeline – overview

Original continuous signal:
\( u : \mathbb{R} \to \mathbb{R} \)

Sampled signal:
\( S : \mathbb{Z} \to \mathbb{R} : n \mapsto u(n) \)

Reconstructed signal:
\( \tilde{S} : \mathbb{R} \to \mathbb{R} \)
(many reconstruction methods)
Fourier Waveform Synthesis – approximate continuous signal

- A signal can be approximated by summing sine (and cosine) waves of different frequencies, phases, and amplitudes.
- A signal has 2 representations. We’re familiar with the **spatial domain**, but every signal also exists in the **frequency domain**.

Approximation of scan line from image improves with more sine waves.
Matlab demo

```matlab
x = [0:0.001:1];
sin0 = sin(2 * pi * 50 * x);
sin1 = sin(2 * pi * 120 * x);
y = 0.7 * sin0 + sin1;
plot(sin0, 'g-'); hold on;
plot(sin1, 'b-'); plot(y, 'r-');

fy = abs(fft(y));
plot(fy(1:size(fy,2)/2));
```

```matlab
sharp = imread('sharp.png');
blurred = imread('blurred.png');
figure
imshow(sharp);
figure
imshow(blurred);

sharp = rgb2gray(sharp);
blurred = rgb2gray(blurred);

fsharp = fft(sharp(40, :));
fbblurred = fft(blurred(40, :));

figure
plot(abs(fbblurred(1:size(fbblurred,2)/2)), 'r-')
hold on
plot(abs(fsharp(1:size(fsharp,2)/2)));
```
Frequency Spectrum of a Signal

- Sine wave is characterized by amplitude and frequency.
- Frequency of a sine wave is number of cycles per second for audio, or number of cycles per unit length (e.g., inter-pixel distance) for image scan lines.
- Can characterize any waveform by enumerating amplitude and frequency of all its component sine waves (Fourier transform – see chapter 18 in book).
- This can be plotted as a “frequency spectrum”, a.k.a. power spectrum, (we ignore negative frequencies, but they are needed for mathematical correctness).

To see spatial and frequency domains of simple signals: [http://www.cs.brown.edu/exploratories/freeSoftware/repository/edu/brown/cs/exploratories/applets/fft1DApp/1d_fast_fourier_transform_guide.html](http://www.cs.brown.edu/exploratories/freeSoftware/repository/edu/brown/cs/exploratories/applets/fft1DApp/1d_fast_fourier_transform_guide.html)
Simple Fourier Synthesis example (1/2)

example: \( g(t) = \sin(2\pi f t) + \frac{1}{3}\sin(2\pi (3f) t) \)
Simple Fourier Synthesis example (2/2)

- Approximate signal using sines and cosines

We are just summing waves to get bolded blue function to approximate the red signal.
Sampling: The Nyquist Limit
Sampling: The Nyquist Limit
Sampling: The Nyquist Limit

- Sampling frequency must be 2 times more than the highest frequency in the signal (the Nyquist limit).
Temporal Aliasing
Temporal Aliasing: Another Sampling Error

- Ever seen tires spin in a movie? Have you ever noticed that sometimes, they seem to be spinning backwards?

- It’s because the video frame-rate is lower than twice the frequency at which the wheels spin. This is temporal aliasing!

- You see this a lot in movies because the effect is so striking. It’s known as the stage-coach effect.

- [Porsche Dyno Test](#)
What’s the issue?

If your content has high frequency components but your display is lower resolution (need to show in a smaller image)
Scale Aliasing, or “Why do we have to pre-filter?”

This doesn’t look right at all. There are no stripes and the image now has a blacker average intensity
Scale Aliasing II, or “Close, but no cigar?”

Original Image

Prefiltered image with samples marked

Prefiltered image scaled

Better, but not perfect...
Solution for anti-aliasing

- Blur an image to kill high-frequency component
- Then, do point-sampling
- This works because high-frequency component becomes even higher in a smaller (low resolution) image,
Scale Aliasing II, or “Close, but no cigar?”

Better, but not perfect...
Simply take every other rows/cols
Blur then take very other rows/cols
Blur by Convolution

Image

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>80</td>
<td>70</td>
<td>40</td>
<td>50</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
<td>80</td>
<td>120</td>
<td>200</td>
<td>180</td>
<td>130</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>230</td>
<td>200</td>
<td>180</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>180</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>10</td>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>160</td>
<td>150</td>
<td>130</td>
<td>180</td>
<td>200</td>
<td>190</td>
<td>150</td>
</tr>
<tr>
<td>30</td>
<td>80</td>
<td>90</td>
<td>100</td>
<td>80</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>

Correlation output

62
### Convolution

#### Image

<table>
<thead>
<tr>
<th></th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>230</th>
<th>20</th>
<th>160</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>305</td>
<td>405</td>
<td>50</td>
<td>50</td>
<td>20</td>
<td>80</td>
<td>30</td>
</tr>
<tr>
<td>40</td>
<td>401</td>
<td>604</td>
<td>80</td>
<td>30</td>
<td>10</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>50</td>
<td>505</td>
<td>705</td>
<td>180</td>
<td>20</td>
<td>80</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>230</td>
<td>200</td>
<td>180</td>
<td>180</td>
<td>40</td>
<td>10</td>
<td>20</td>
<td>80</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>10</td>
<td>80</td>
<td>180</td>
<td>180</td>
</tr>
<tr>
<td>160</td>
<td>150</td>
<td>130</td>
<td>180</td>
<td>200</td>
<td>200</td>
<td>190</td>
<td>190</td>
</tr>
<tr>
<td>30</td>
<td>80</td>
<td>90</td>
<td>100</td>
<td>80</td>
<td>20</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

#### Correlation output

| 62 | 71 |

---

Courtesy of Andries van Dam©
### Convolution

#### Image

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>80</td>
<td>40</td>
<td>90</td>
<td>40</td>
<td>60</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
<td>80</td>
<td>40</td>
<td>60</td>
<td>40</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>50</td>
<td>90</td>
<td>40</td>
<td>60</td>
</tr>
<tr>
<td>230</td>
<td>200</td>
<td>180</td>
<td>30</td>
<td>40</td>
<td>60</td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>10</td>
<td>80</td>
</tr>
<tr>
<td>160</td>
<td>150</td>
<td>130</td>
<td>180</td>
<td>200</td>
<td>190</td>
</tr>
<tr>
<td>30</td>
<td>80</td>
<td>90</td>
<td>100</td>
<td>80</td>
<td>20</td>
</tr>
</tbody>
</table>

#### Correlation output

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>62</td>
<td>71</td>
<td>102</td>
</tr>
</tbody>
</table>
Convolution

Image

<table>
<thead>
<tr>
<th></th>
<th>30</th>
<th>80</th>
<th>70</th>
<th>40</th>
<th>50</th>
<th>50</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>40</td>
<td>80</td>
<td>120</td>
<td>200</td>
<td>130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>70</td>
<td>80</td>
<td>90</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>230</td>
<td>200</td>
<td>180</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>180</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>30</td>
<td>40</td>
<td>50</td>
<td>10</td>
<td>80</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>150</td>
<td>130</td>
<td>180</td>
<td>200</td>
<td>190</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>80</td>
<td>90</td>
<td>100</td>
<td>80</td>
<td>20</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

Correlation output

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>62</th>
<th>71</th>
<th>102</th>
<th>98</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Correlation (review)

(image)

(Values in the output are fake.)
Convolution for edge detection

Gradient along X

Gradient along Y
Convolution with 

\[
\begin{bmatrix}
-0.5 & 0 & 0.5
\end{bmatrix}
\]

Convolution with 

\[
\begin{bmatrix}
0.5 \\
0 \\
0.5
\end{bmatrix}
\]
Better kernels/filters

\[ h_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \]

Gaussian

\[ \frac{\partial}{\partial x} h_\sigma(x, y) \]

derivative of Gaussian
Low-Pass Filtering to Eliminate High

Fig. 14.20 The sampling pipeline with filtering. (Courtesy of George Wolberg, Columbia University.)
Convolving signal \( f(x) \) with filter function \( g(x) \):

\[
h(x) = f(x) * g(x) = \int_{-\infty}^{+\infty} f(\tau) g(x - \tau) d\tau
\]

- At each point \( x \), \( h(x) \) is integral of product of \( f \) and \( g \), except \( g \) is flipped and translated so its origin is at \( x \)
  - In practice, \( f \) or \( g \) is often an even function (symmetric about the y-axis) and we don’t need to flip
- If \( g(x) \) has finite support, it does a weighted average centered at \( x \)
- \( f(x) \) (blue signal) convolved by \( g(x) \) (red filter) to get result \( h(x) \) (black signal)
  - \( f(x) \) and \( g(x) \) are box functions
  - Each point on black signal is result of an integral, represented by the area under the product of \( f(x) \) and \( g(x) \) (yellow area)
Simple Convolution Example

Convolution is a lot like multiplication
\[ f(x) \ast f(x) \]

\[
\begin{array}{c}
1111 \\
\times 1111 \\
1111 \\
1111 \\
+ 1111 \\
\hline
1234321
\end{array}
\]
Ideal Low-Pass Filtering (Frequency Domain)

- Multiplying by the *box* function in the frequency domain
- Frequencies under the box are kept, and the high frequencies are cut off
- Corresponds to convolution with the *sinc* function in the spatial domain

\[ sinc(x) = \frac{\sin(\pi x)}{\pi x} \]
Common Filters and Their Duals

Spatial domain

Box
Triangle
Finite Gaussian

Frequency domain

Sinc
Sinc²
Gaussian

Bad filter in practice: gradual attenuation, negative lobes, infinite extent, all corrupting signal. But still beats point sampling!
In theory sinc has infinite extent, however small the contributions and negative lobes, but weights contributions at the center most heavily.

Practically, sinc is decently approximated with gaussian (normal) distribution, or even triangle, with finite extents and weights greater than or equal to zero.
Properties of Convolution

- Commutative
  \[ f * g = g * f \]
- Associative
  \[ f * (g * h) = (f * g) * h \]
- Distributive
  \[ f * (g + h) = (f * g) + (f * h) \]
- Identity
  \[ f * \delta = f \]

\( \delta \) is the Dirac delta— a mathematical artifice

- 0 everywhere except at \( \delta(0) = \infty \)
- Area under \( \delta \) is 1

\[
\int \delta(x) dx = 1
\]

\( \delta(x) = \lim_{a \to 0} \frac{1}{a\sqrt{\pi}} e^{-x^2/a^2} \)
Approximation

Sinc in spatial domain corresponds to box/pulse in frequency domain

Truncated sinc in spatial domain corresponds to ringing pulse in frequency domain – decent approximation to perfect pulse
## Duals

<table>
<thead>
<tr>
<th>Spatial</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplication</td>
<td>Convolution</td>
</tr>
<tr>
<td>Convolution</td>
<td>Multiplication</td>
</tr>
<tr>
<td>Box</td>
<td>Sinc</td>
</tr>
<tr>
<td>Sinc</td>
<td>Box</td>
</tr>
</tbody>
</table>