Review
- Images – an array of colors
- Color – RGBA
- Loading, modifying, updating pixels
- pixels[] as a 2D array
- Simple filters – tinting, grayscale, negative, sepia
- PImage class, fields and methods
- get() method and crumble
- tint() function – color and alpha filtering
- Creative image processing – Pointillism, other shapes

Medical Images

Image Processing in Manufacturing

What can you do with Image Processing?
- Inspect, Measure, and Count using Photos and Video
  http://www.youtube.com/watch?v=86sTNNWVhpg
- Image Processing Software
  http://www.youtube.com/watch?v=tWUp9mGnWSM

Thresholding for Image Segmentation
- Pixels below a cutoff value are set to black
- Pixels above a cutoff value are set to white

Obamicon
Example
- obamicon

Image Enhancement
- Color and intensity adjustment
- Histogram equalization

Histogram Equalization
- Increases the global contrast of images
- So that intensities are better distributed
- Reveals more details in photos that are over or under exposed
- Better views of bone structure in X-rays

Histogram Equalization
- Calculate color frequencies - count the number of times each pixel color appear in the image
- Calculate the cumulative distribution function (cdf) for each pixel color – the number of times all smaller color values appear in the image
- Normalize over (0, 255)

Convolution Filters (Area-based)
\[ E' = w_1A + w_2B + w_3C + w_4D + w_5E + w_6F + w_7G + w_8H + w_9I \]
Identity
- No change

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Random Neighbor
- Copies randomly from one of the 8 neighbors, and itself

Example
- randomNeighbor

Average – smooth
- Set pixel to the average of all colors in the neighborhood
- Smoothes out areas of sharp changes.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
</tr>
<tr>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
</tr>
<tr>
<td>1/9</td>
<td>1/9</td>
<td>1/9</td>
</tr>
</tbody>
</table>

Sharpen – High Pass Filter
- Enhances the difference between neighboring pixels
- The greater the difference, the more change in the current pixel

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>-2/3</td>
</tr>
<tr>
<td>-1</td>
<td>9</td>
<td>-1</td>
<td>-2/3</td>
<td>11/3</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>-2/3</td>
</tr>
</tbody>
</table>

Blur – Low Pass Filter
- Softens significant color changes in image
- Creates intermediate colors

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1/16</td>
<td>2/16</td>
<td>1/16</td>
</tr>
<tr>
<td>2/16</td>
<td>4/16</td>
<td>2/16</td>
</tr>
<tr>
<td>1/16</td>
<td>2/16</td>
<td>1/16</td>
</tr>
</tbody>
</table>
Example

- convolution

Dilation - Morphology

- Set pixel to the maximum color value within a neighborhood around the pixel
- Causes objects to grow in size.
- Brightens and fills in small holes

Erosion - Morphology

- Set pixel to the minimum color value within a neighborhood around the pixel
- Causes objects to shrink.
- Darkens and removes small objects

Feature Extraction - Region Detection

- Dilate and Erode
  - Open
    - Erode → dilate
    - Removes noise
  - Close
    - Dilate → Erode
    - Holes are closed

Erode + Dilate to Despeckle

Image Enhancement

- Denoise
  - Averaging
  - Median filter
**Image Processing in Processing**

- `tint()`: modulate individual color components
- `blend()`: combine the pixels of two images in a given manner
- `filter()`: apply an image processing algorithm to an image

**Blend Command**
```
img = loadImage("colony.jpg");
mask = loadImage("mask.png");
image(img, 0, 0);
blend(img, 0, 0, mask.width, mask.height, 0, 0, img.width, img.height, SUBTRACT);
```

**BLEND linear interpolation of colours:**
```
C = A*factor + B
```

**ADD additive blending with white clip:**
```
C = min(A*factor + B, 255)
```

**SUBTRACT subtractive blending with black clip:**
```
C = max(A*factor - B, 0)
```

**DARKEST only the darkest colour succeeds:**
```
C = max(A*factor, B)
```

**LIGHTEST only the lightest colour succeeds:**
```
C = min(A*factor, B)
```

**DIFFERENCE subtract colors from underlying image.**

**EXCLUSION similar to DIFFERENCE, but less extreme.**

**MULTIPLY Multiply the colors, result will always be darker.**

**SCREEN Opposite multiply, uses inverse values of the colors.**

**OVERLAY Mix of MULTIPLY and SCREEN. Multiples dark values, and screens light values.**

**HARD_LIGHT SCREEN when greater than 50% gray, MULTIPLY when lower.**

**SOFT_LIGHT Mix of DARKEST and LIGHTEST. Works like OVERLAY, but not as harsh.**

**DODGE Lightens light tones and increases contrast, ignores darks.**

**BURN Darker areas are applied, increasing contrast, ignores lights.**

**Filter Command**
```
// Threshold
PImage img;
void setup() {
  img = loadImage("myImage.jpg");
  size(img.width, img.height);
  image(img, 0, 0);
}
void draw() {
  filter(THRESHOLD, 0.5);
}
```

**THRESHOLD converts the image to black and white pixels depending if they are above or below the threshold defined by the level parameter.**

**GRAY converts any colors in the image to grayscale equivalents.**

**INVERT sets each pixel to its inverse value.**

**POSTERIZE limits each channel of the image to the number of colors specified as the level parameter.**

**BLUR executes a Gaussian blur with the level parameter specifying the extent of the blurring.**

**OPAQUE sets the alpha channel to entirely opaque.**

**ERODE reduces the light areas with the amount defined by the level parameter.**

**DILATE increases the light areas with the amount defined by the level parameter.**

**Image Processing Applications**

**Manual Colony Counter**
```
http://www.youtube.com/watch?v=7B-9Wf6pENQ
```

---

![Image Processing Examples](image.png)
Measuring Confluency in Cell Culture Biology

- Refers to the coverage of a dish or flask by the cells
- 100% confluency = completely covered

- Image Processing Method
  1. Mask off unimportant parts of image
  2. Threshold image
  3. Count pixels of certain color

Filter: Threshold

Count pixels to quantify: 5.3% confluency

Blend: Subtract

Vision Guided Robotics Colony Picking

Predator algorithm for object tracking with learning
http://www.youtube.com/watch?v=1GhNXHCQGsM

Video Processing, with Processing
http://www.youtube.com/watch?v=rKhbUjVyKlc