Internet of Things
Sensors - Part 3
Image and Speech Processing

Aveek Dutta
Assistant Professor
Electrical Engineering and Computer Science
University of Kansas
e-mail: aveekd@ku.edu
http://www.ittc.ku.edu/~aveekd
Image Transformation

- Per-pixel transformation
  - Histogram
  - Linear Filtering
- Edges, corners, and interest points
- Geometry, Pose and orientation
Whitening

- Invariant to fluctuations in the mean intensity level and contrast of the image
  - Due to ambient lighting, camera gain, object reflectance
- Image is transformed to have zero mean and unit invariance of pixel brightness

\[
\mu = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J} p_{ij}}{IJ}
\]

\[
\sigma^2 = \frac{\sum_{i=1}^{I} \sum_{j=1}^{J} (p_{ij} - \mu)^2}{IJ}
\]

- For color images use separate RGB channels

Content adapted from Computer Vision: Models, Learning, and Inference by Simon J.D. Prince (http://www.computervisionmodels.com/)
Histogram Equalization

- Modify statistics of the intensity values so that all moments take predefined values
- Histogram of the original intensities $h$ where the $k^{th}$ of $K$ entries is given by

$$h_k = \sum_{i=1}^{I} \sum_{j=1}^{J} \delta[p_{ij} - k]$$

$$c_k = \frac{\sum_{l=1}^{k} h_{l}}{IJ}$$

$$x_{i,j} = Kc_{p_{ij}}$$

Content adapted from Computer Vision: Models, Learning, and Inference by Simon J. D. Prince (http://www.computervisionmodels.com/)
Linear filtering

- Same as 1D signal transformation but in 2-dimensions
- Basic operation - new pixel value $x_{ij}$ as a weighted sum of surrounding pixels
  - Weights are nothing but filter coefficients (or kernel) $F$
- What operation is used to filter? aka, flipped cross-correlation

$$ (f * g_N)[n] = \sum_{m=0}^{N-1} f[m] g_N[n - m] $$

![Convolution, Cross-correlation, Autocorrelation](http://creativecommons.org/licenses/by-sa/3.0)
2D Convolution

- Convolve pixels $p$ with filter $f$

$$x_{ij} = \sum_{m=-M}^{M} \sum_{n=-N}^{N} p_{i-m,j-n} f_{m,n}$$

- By convention, the filter is flipped in both directions so the top left of the filter $f_{-M,-N}$ weights the pixel $p_{i+M,j+N}$ to the right and below the current point in $P$.
Blurring as a filter

Each pixel in the resulting image is a weighted sum of the surrounding pixels, where the weights depend on the Gaussian profile: nearer pixels contribute relatively to the final output as per the following equation.

\[
f(m, n) = \frac{1}{2\pi\sigma^2} \exp \left[ -\frac{m^2 + n^2}{2\sigma^2} \right]
\]

Content adapted from Computer Vision: Models, Learning, and Inference by Simon J.D. Prince (http://www.computervisionmodels.com/)
Edge Detection and Reconstruction

- Filters are linear operators - Edges preserve the brightness information on either side of the edge. Reverse operation (almost) reconstructs the original image.
Edge filters

Original image

Prewitt (vertical)

\[
\begin{bmatrix}
1 & 1 & 1 \\
0 & 0 & 0 \\
-1 & -1 & -1 \\
\end{bmatrix}
\]

Prewitt (horizontal)

\[
\begin{bmatrix}
1 & 0 & -1 \\
1 & 0 & -1 \\
1 & 0 & -1 \\
\end{bmatrix}
\]

Content adapted from Computer Vision: Models, Learning, and Inference by Simon J.D. Prince (http://www.computervisionmodels.com/)
Edge Detection

- Edge = Jump (discontinuity) of brightness (intensity)
- Filters useless information but preserves structural properties in an image
- Gradient Method in one dimensional signal
Sobel Edge Detection

- Sobel operator performs a 2D spatial gradient using a pair of filters

\[
|G| = \sqrt{Gx^2 + Gy^2}
\]

<table>
<thead>
<tr>
<th>Gx</th>
<th>Gy</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
b_{22} = (a_{11} \times m_{11}) + (a_{12} \times m_{12}) + (a_{13} \times m_{13}) + (a_{21} \times m_{21}) + (a_{22} \times m_{22}) + (a_{23} \times m_{23}) + (a_{31} \times m_{31}) + (a_{32} \times m_{32}) + (a_{33} \times m_{33})
\]
Canny Edge Detector

1. Blur (using Gaussian filters)

2. Apply differentiation along X and Y axes (Prewitt, Sobel) gives $|G| = |G_x| + |G_y|$

3. Calculate gradient orientation $\theta = \arctan \left( \frac{G_y}{G_x} \right)$ and quantize

4. Non-maximum suppression: Set any pixel value equal to 0 if either of the neighboring two pixels perpendicular to the gradient have higher values

5. Hysteresis to eliminate streaking.
   a. Any pixel value greater than T1 is an edge pixel.
   b. Any connected with value than T2 are also marked as edge pixels.
Depth, Orientation and Pose

Sparse Stereo Reconstruction
Pinhole Camera Model

Account for photo-perceptor spacing, offset and skew

\[ x = \frac{\phi_x u}{w} + \delta_x \]
\[ y = \frac{\phi_y v}{w} + \delta_y \]
Orientation

Where \( w' \) is the transformed point, \( \Omega \) is a \( 3 \times 3 \) rotation matrix, and \( \tau \) is a \( 3 \times 1 \) translation vector.

\[ \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{21} & \omega_{22} & \omega_{23} \\ \omega_{31} & \omega_{32} & \omega_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}, \]

\[ w' = \Omega w + \tau, \]

- \( \Omega \) and \( \tau \) are extrinsic parameters and rest (\( u,v,w \), others) are intrinsic.
- The generative pinhole model describes the likelihood \( \Pr(x|w, \Lambda, \Omega, \tau) \) of observing a 2D image point \( x \) given a 3D world point \( w \) and the parameters \( \{ \Lambda, \Omega, \tau \} \).

\[ \Pr(x|w, \Lambda, \Omega, \tau) = \text{Norm}_x \left[ \text{pinhole}[w, \Lambda, \Omega, \tau], \sigma^2 I \right] \]