Lecture 7 – Knowing a Good Feature, Feature Matching, and Recognition

Ulas Bagci

bagci@ucf.edu
Outline

• What is good feature?
• Image Value Analysis
  – Haralick
• Line Detection
  – Hough Transform

Reminder: Select your project#1 by 15 October 2016

Slides Credits: Dr. Shah and Dr. Szeliski
Computer Vision Applications

Image-Feature Extraction  Machine Learning/Pattern Analysis  Classification, Segmentation, Detection, Tracking, ....
Local Features

Building blocks of many visual recognition systems

Designed for the specific tasks, exploiting common sense

Deterministic functions of the image designed to represent the object
Local Features

Building blocks of many visual recognition systems

How to evaluate effectiveness of local features?

Designed for the specific tasks, exploiting common sense

Deterministic functions of the image designed to represent the object
Local Features

Building blocks of many visual recognition systems

How to evaluate effectiveness of local features?

- Affine invariance
- Robustness
- Accuracy in specific tasks (tracking, recognition, localization, detection,..)

Detected for the specific tasks, exploiting common sense

Deterministic functions of the image designed to represent the object
Pyramid Representation - Scale Info.

What is a good feature?
Invariance in geometry and illumination is desired. What else is required for a good feature?

- Template-based (i.e., patch) methods (e.g., conv, corr, ..) lack robustness to transformation
- Histogram-based methods (e.g., SIFT) have so much flexibility that discrimination can be degraded.
Invariance in geometry and illumination is desired. What else is required for a good feature?

- Template-based (i.e., patch) methods (e.g., conv, corr, ..) lack robustness to transformation
- Histogram-based methods (e.g., SIFT) have so much flexibility that discrimination can be degraded.

Image Value Analysis!
Image Value Analysis in Recognition

• describes the given signal, i.e., the distribution of image values
Image Value Analysis in Recognition

• describes the given signal, i.e., the distribution of image values
• Basic statistics (mean, variance, grey-level histogram) provide measures summarizing individual pixel values
Image Value Analysis in Recognition

• describes the given signal, i.e., the distribution of image values

• Basic statistics (mean, variance, grey-level histogram) provide measures summarizing individual pixel values

• Co-occurrence Matrices and Measures – study the distribution of values in dependence upon values at adjacent pixels
Image Value Analysis in Recognition

• describes the given signal, i.e., the distribution of image values

• Basic statistics (mean, variance, grey-level histogram) provide measures summarizing individual pixel values

• Co-occurrence Matrices and Measures
  – study the distribution of values in dependence upon values at adjacent pixels
  – are represented in the co-occurrence matrix C

HARALICK and SHAPIRO, 1992.
Co-occurrence Matrix

• Assume an input image $I$, and adjacency set $A$. For example, in case of 1,2,3,4-adjacency, we have the adjacency set

$$A_1 = \{(0, 1)\}$$

$$A_2 = \{(0, 1), (1, 0)\}$$

$$A_3 = \{(0, 1), (1, 0), (0, -1)\}$$

$$A_4 = \{(0, 1), (1, 0), (0, -1), (-1, 0)\}$$
Co-occurrence Matrix

• Assume an input image I, and adjacency set A. For example, in case of 4-adjacency, we have the adjacency set

\[ A_4 = \{(0, 1), (1, 0), (0, -1), (-1, 0)\} \]

\[ A_4(p) = A_4 + p \]
Co-occurrence Matrix

• We denote by $\Omega$ the set of all $N_{cols} \times N_{rows}$ pixel locations.
Co-occurrence Matrix

- We denote by $\Omega$ the set of all $N_{cols} \times N_{rows}$ pixel locations.
- We define the $(G_{max}+1) \times (G_{max}+1)$ co-occurrence matrix $C_i$ for image $I$ and image values $u$ and $v$ in $\{0,1,...,G_{max}\}$ as follows:

$$C_I(u, v) = \sum_{p \in \Omega, q \in \Omega} \sum_{p+q \in \Omega} \begin{cases} 1 & \text{if } I(p) = u \text{ and } I(p + q) = v \\ 0 & \text{otherwise.} \end{cases}$$
## Co-occurrence Matrix

### 4 x 4 co-occurrence matrix

Only A={(0,1)} is demonstrated.

(5,1) \rightarrow go to 0,0 increase by 1

**Gmax=3**

Smallest value=0

5 x 5 image
Co-occurrence Matrix

\[
\begin{array}{cccc}
2 & 1 & 1 & 0 & 0 \\
3 & 3 & 2 & 2 & 0 \\
3 & 3 & 1 & 2 & 0 \\
2 & 3 & 1 & 1 & 0 \\
3 & 3 & 2 & 1 & 1 \\
\end{array}
\]

\[
\begin{array}{ccc}
& u & C_1 \\
3 & 0 & 0 & 0 \\
1 & 2 & 2 & 0 \\
1 & 2 & 1 & 1 \\
0 & 1 & 2 & 4 \\
\end{array}
\]

\[
\begin{array}{ccc}
& u & C_2 \\
4 & 2 & 2 & 0 \\
1 & 5 & 4 & 2 \\
1 & 3 & 2 & 3 \\
0 & 1 & 3 & 7 \\
\end{array}
\]

\[
\begin{array}{ccc}
& u & C_3 \\
7 & 3 & 3 & 0 \\
1 & 7 & 6 & 3 \\
1 & 5 & 3 & 5 \\
0 & 1 & 4 & 11 \\
\end{array}
\]

\[
\begin{array}{ccc}
& u & C_4 \\
8 & 3 & 3 & 0 \\
3 & 10 & 7 & 3 \\
3 & 7 & 4 & 6 \\
0 & 3 & 6 & 14 \\
\end{array}
\]
Co-occurrence Matrix

\[ A_1 = \{(0, 1)\} \]
Co-occurrence Matrix

\[ A_1 = \{(0, 1)\} \]
**Co-occurrence Matrix**

\[ A_1 = \{(0, 1)\} \]
Co-occurrence Matrix

$A_1 = \{(0, 1)\}$
Co-occurrence Matrix

\[ A_2 = \{(0, 1), (1, 0)\} \]
Co-occurrence Matrix

\[
A_2 = \{(0, 1), (1, 0)\}
\]
Co-occurrence based measures

• are used to quantify information in the image
Co-occurrence based *measures*

- are used to quantify information in the image.
- Note that noise in an image is still considered to be information when using these measures.
Co-occurrence based measures

• are used to quantify information in the image.
• Note that noise in an image is still considered to be information when using these measures.
• We provide here two of such measures:
Co-occurrence based measures

- are used to quantify information in the image
- Note that noise in an image is still considered to be information when using these measures.
- We provide here two of such measures:

\[
M_{\text{hom}}(I) = \sum_{u,v \in \{0,1,\ldots,G_{\text{max}}\}} \frac{C_I(u, v)}{1 + |u - v|}
\]
Co-occurrence based measures

- are used to quantify information in the image $I$
- Note that noise in an image is still considered to be information when using these measures.
- We provide here two of such measures:

\[ M_{hom}(I) = \sum_{u,v \in \{0,1,\ldots,G_{max}\}} \frac{C_I(u, v)}{1 + |u - v|} \]  

\[ M_{uni}(I) = \sum_{u,v \in \{0,1,\ldots,G_{max}\}} C_I(u, v)^2 \]

(Homogeneity)

(Uniform)
Moment-Based Region Analysis

• Assume a region \( S \subset \Omega \) of pixel locations in an image.
Moment-Based Region Analysis

- Assume a region $\mathcal{S} \subset \Omega$ of pixel locations in an image $I$
- This region may represent an object such as
Moment-Based Region Analysis

- Assume a region $S \subset \Omega$ of pixel locations in an image $I$
- This region may represent an object such as
- We assume again a scalar image $I$. The moments of a region $S$ in the image $I$ are defined by
Moment-Based Region Analysis

• Assume a region $S \subset \Omega$ of pixel locations in an image $I$
• This region may represent an object such as
• We assume again a scalar image $I$. The moments of a region $S$ in the image $I$ are defined by

$$m_{a,b}(S) = \sum_{(x,y) \in S} x^a y^b I(x, y)$$

For non-negative integers $a$ and $b$. The sum $a+b$ defines the Order of the moment. There is only one moment of order zero
Moment-Based Region Analysis

• Assume a region $S \subset \Omega$ of pixel locations in an image $I$
• This region may represent an object such as
• We assume again a scalar image $I$. The moments of a region $S$ in the image $I$ are defined by

$$m_{a,b}(S) = \sum_{(x,y) \in S} x^a y^b I(x, y)$$

For non-negative integers $a$ and $b$. The sum $a+b$ defines the Order of the moment. **There is only one moment of order zero**

$$m_{0,0}(S) = \sum_{(x,y) \in S} I(x, y)$$
Moment-Based Region Analysis

- Assume a region $S \subset \Omega$ of pixel locations in an image $I$
- This region may represent an object such as
- We assume again a scalar image $I$. The moments of a region $S$ in the image $I$ are defined by

$$m_{a,b}(S) = \sum_{(x,y) \in S} x^a y^b . I(x, y)$$

For non-negative integers $a$ and $b$. The sum $a+b$ defines the Order of the moment. The moments of order 1:

$$m_{1,0}(S) = \sum_{(x,y) \in S} x . I(x, y) \quad m_{0,1}(S) = \sum_{(x,y) \in S} y . I(x, y)$$
Centroids and Central Moments

Centroids:

\[ x_S = \frac{m_{1,0}(S)}{m_{0,0}(S)} \quad \text{and} \quad y_S = \frac{m_{0,1}(S)}{m_{0,0}(S)} \]
Centroids and Central Moments

**Centroids:**

\[ x_S = \frac{m_{1,0}(S)}{m_{0,0}(S)} \quad y_S = \frac{m_{0,1}(S)}{m_{0,0}(S)} \]

**Central Moments:**

\[ \mu_{a,b}(S) = \sum_{(x,y) \in S} (x - x_S)^a (y - y_S)^b \cdot I(x, y) \]
Centroids and Central Moments

Centroids:

\[ x_S = \frac{m_{1,0}(S)}{m_{0,0}(S)} \quad y_S = \frac{m_{0,1}(S)}{m_{0,0}(S)} \]

Central Moments:

\[ \mu_{a,b}(S) = \sum_{(x,y) \in S} (x - x_S)^a (y - y_S)^b \cdot I(x, y) \]

The central moments provide a way to characterize regions S by features that are invariant with respect to any linear transform!
Detection of Lines

• Real-world images often show straight lines, such as edges of buildings, lane borders, or power poles
Detection of Lines

- Real-world images often show straight lines, such as edges of buildings, lane borders, or power poles.
Detection of Lines

• Real-world images often show straight lines, such as edges of buildings, lane borders, or power poles
  – An edge detector is first used to map a given image into a binary edge map
Detection of Lines

- Real-world images often show straight lines, such as edges of buildings, lane borders, or power poles
  - An edge detector is first used to map a given image into a binary edge map
  - Then, all remaining edge pixels need to be analyzed whether there is any subset that forms “reasonably” a line segment
Detection of Lines

- Real-world images often show straight \textit{lines}, such as edges of buildings, lane borders, or power poles
- An edge detector is first used to map a given image into a binary edge map
- Then, all remaining edge pixels need to be analyzed whether there is any subset that forms “reasonably” a line segment
- There is noise involved, and the line detector needs to be robust with respect to some noise
Detection of Lines

• Real-world images often show straight lines, such as edges of buildings, lane borders, or power poles
  – An edge detector is first used to map a given image into a binary edge map
  – Then, all remaining edge pixels need to be analyzed whether there is any subset that forms “reasonably” a line segment
  – There is noise involved, and the line detector needs to be robust with respect to some noise

• How to describe a line?
Detection of Lines

• Real-world images often show straight lines, such as edges of buildings, lane borders, or power poles
  – An edge detector is first used to map a given image into a binary edge map
  – Then, all remaining edge pixels need to be analyzed whether there is any subset that forms “reasonably” a line segment
  – There is noise involved, and the line detector needs to be robust with respect to some noise

• How to describe a line?

\[ y = a \cdot x + b \]
Hough Transform

• Connection between image (x,y) and Hough (m,b) spaces
  – A line in the image corresponds to a point in Hough space
  – To go from image space to Hough space:
    • given a set of points (x,y), find all (m,b) such that $y = mx + b$
Hough Transform

What does a point \((x_0, y_0)\) in the image space map to?

- A: the solutions of \(b = -x_0m + y_0\)
- this is a line in Hough space
Original Hough Transform
Pixel positions $p, q,$ and $r$ in the image are mapped into three lines in the parameter space.
Pixel positions $p, q,$ and $r$ in the image are mapped into three lines in the parameter space. $y=a_1.x+b_1$ (blue) $\implies b=-xp.a + yp$ and $b=-xq.a+yq$
Original Hough Transform

Pixel positions $p, q,$ and $r$ in the image are mapped into three lines in the parameter space.

$y = a_1 x + b_1$ (blue) --> $b = -xp.a + yp$ and $b = -xq.a + yq$

However, since $a$ and $b$ are not bounded by $Ncols \times Nrows$ image, infinite parameter space will not allow detecting lines in practice.
Parameter Space by Duda and Hart - Line Detection

- Angular parameterization can help bounding the parameter space

\[ d = x \cdot \cos \alpha + y \cdot \sin \alpha \]
Parameter Space by Duda and Hart - Line Detection

- Angular parameterization can help bounding the parameter space
  
  \[ d = x \cdot \cos \alpha + y \cdot \sin \alpha \]

- Angular parameter alpha is in the interval of \([0, 2\pi)\), and the distance \(d\) is in the interval of \([0, d_{\text{max}}]\) with

\[ d_{\text{max}} = \sqrt{(N_{\text{cols}}^2 + N_{\text{rows}})} \]
Parameter Space by Duda and Hart-Line Detection

- Angular parameterization can help bounding the parameter space

\[ d = x \cdot \cos \alpha + y \cdot \sin \alpha \]

- Angular parameter \( \alpha \) is in the interval of \([0,2\pi)\), and the distance \( d \) is in the interval of \([0,d_{\text{max}}]\) with

\[ d_{\text{max}} = \sqrt{N_{\text{cols}}^2 + N_{\text{rows}}} \]

- A point in the image generates now a sin/cos curve in the \( d-\alpha \) parameter space, also known as **Houghspace**
Parameter Space by Duda and Hart - Line Detection
Hough Space as Discrete Accumulator Array

- For implementing Hough space, it is digitized into an array, using subsequent intervals $d$ and alpha for defining a finite number of cells in Hough space.
Hough Space as Discrete Accumulator Array

- For implementing Hough space, it is digitized into an array, using subsequent intervals $d$ and alpha for defining a finite number of cells in Hough space.
- $d \rightarrow [0,1), [1,2), [2,3)\ldots$ accumulator array
Hough Space as Discrete Accumulator Array

• For implementing Hough space, it is digitized into an array, using subsequent intervals $d$ and alpha for defining a finite number of cells in Hough space.

  $d \rightarrow [0,1), [1,2), [2,3)....accumulator\ array$

• Set a counter for each cell and increase it by one for each sin/cos curve passing through that cell
Hough Transform Pseudo-Code

Create \( \theta \) and \( d \) for all possible lines
Create an array \( A \) indexed by \( \theta \) and \( d \)

\[
\text{for each point } (x,y) \\
\quad \text{for each angle } \phi \\
\quad \quad d = x \cos(\theta) + y \sin(\theta) \\
\quad A[\theta, d] = A[\theta, d] + 1 \\
\text{end} \\
\text{end}
\]

where \( A > \text{Threshold} \) return a line
A Simple Example
A Simple Example 2
A Real World Example

Parameter space

edges
Hough Space as Discrete Accumulator Array

3x3 max sum

Butterfly of a peak
### Generalized Hough Transform

<table>
<thead>
<tr>
<th>Analytic Form</th>
<th>Parameters</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>$\rho, \theta$</td>
<td>$x \cdot \cos \theta + y \cdot \sin \theta = \rho$</td>
</tr>
<tr>
<td>Circle</td>
<td>$x_0, y_0, \rho$</td>
<td>$(x-x_0)^2 + (y-y_0)^2 = \rho^2$</td>
</tr>
<tr>
<td>Parabola</td>
<td>$x_0, y_0, \rho, \theta$</td>
<td>$(y-y_0)^2 = 4\rho(x-x_0)$</td>
</tr>
<tr>
<td>Ellipse</td>
<td>$x_0, y_0, a, b, \theta$</td>
<td>$(x-x_0)^2/a^2 + (y-y_0)^2/b^2 = 1$</td>
</tr>
</tbody>
</table>
Generalized Hough Transform

• Advantages
  1. A method for object recognition
  2. Robust to partial deformation in shape
  3. Tolerant to noise
  4. Can detect multiple occurrences of a shape in the same pass

• Disadvantages
  1. Lot of memory and computation is required