CEng 583 - Computational Vision

2011-2012 Spring
Week – 3

12th of March, 2011
# Tentative Schedule:

<table>
<thead>
<tr>
<th>Week &amp; Date</th>
<th>Topic</th>
</tr>
</thead>
</table>
| 1           | **Introduction to Vision.**  
What is vision? What are its goals and problems? What are the main processing stages? |
| 2           | **Low-level Vision.**  
| 3           | **Early Vision.**  
| 4           | **3D Vision.**  
Monocular and binocular cues. 3D reconstruction. |
| 5           | **Applications.**  
| 6           | **Paper presentations with theme:** Monocular depth estimation. |
| 7           | **Paper presentations with theme:** Image annotation. |
| 8           | **Paper presentations with theme:** Object/shape modelling. Object recognition. |
| 9           | **Paper presentations with theme:** Feature Descriptors. |
| 10          | **Paper presentations with theme:** Context. Saliency. Attention. |
| 11          | Project Presentations |
| 12          | Project presentations |
| 13          | Project presentations |
| 14          | Project presentations |
Today

- Early Vision
  - Corners
  - Texture
  - Segmentation
  - Optic flow
Image matching

by Diva Sian

by swashford

Slide: Trevor Darrell
Harder case

by Diva Sian

by scgbt

Slide: Trevor Darrell
Harder still?

NASA Mars Rover images

Slide: Trevor Darrell
Corners, Junctions

* Non-accidental features (Witkin & Tenenbaum, 1983)
Corners or Junctions

Non-accidental features (Witkin & Tenenbaum, 1983)

<table>
<thead>
<tr>
<th>2-D Relation</th>
<th>3-D Inference</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Collinearity of points or lines</td>
<td>Collinearity in 3-Space</td>
<td></td>
</tr>
<tr>
<td>2. Curvilinearity of points of arcs</td>
<td>Curvilinearity in 3-Space</td>
<td></td>
</tr>
<tr>
<td>3. Symmetry (Skew Symmetry ?)</td>
<td>Symmetry in 3-Space</td>
<td></td>
</tr>
<tr>
<td>4. Parallel Curves (Over Small Visual Angles)</td>
<td>Curves are parallel in 3-Space</td>
<td></td>
</tr>
<tr>
<td>5. Vertices -- two or more terminations at a common point</td>
<td>Curves terminate at a common point in 3-Space</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4. Five nonaccidental relations. (From Figure 5.2, Perceptual organization and visual recognition [p. 77] by David Lowe. Unpublished doctoral dissertation, Stanford University. Adapted by permission.)
What is accidental?

http://en.wikipedia.org/wiki/Penrose_triangle
Corners as distinctive interest points

- Shifting a window in any direction should give a large change in intensity.

“flat” region: no change in all directions

“edge”: no change along the edge direction

“corner”: significant change in all directions

Source: A. Efros
We will talk about two widely used corner detectors.

* SUSAN Detector
* Moravec Detector
* Harris Detector
Center pixel is compared with the pixels in a circular mask.

- If they are all the same, the pixel is “homogeneous”
- If half of the pixels are different, the pixel is “edge-like”
- If one-quarter of the pixels are different, then the pixel is corner.
Moravec Detector

* Based on “self-similarity”
* Move a window in horizontal, vertical and diagonal directions.
* Compute the similarity of the original patch with the shifted ones.
* A corner is a local minimum in this similarity space.
Harris Detector formulation

Change of intensity for the shift \([u,v]\): 

\[
E(u,v) = \sum_{x,y} w(x,y) \left[ I(x+u, y+v) - I(x,y) \right]^2
\]

Window function
Shifted intensity
Intensity

Window function \(w(x,y) = \)

1 in window, 0 outside
Gaussian

Source: R. Szeliski
Taylor Series for 2D Functions

\[ f(x+u, y+v) = f(x, y) + uf_x(x, y) + vf_y(x, y) + \]

First partial derivatives

\[ \frac{1}{2!} \left[ u^2 f_{xx}(x, y) + uv f_{xy}(x, y) + v^2 f_{yy}(x, y) \right] + \]

Second partial derivatives

\[ \frac{1}{3!} \left[ u^3 f_{xxx}(x, y) + u^2 v f_{xxy}(x, y) + uv^2 f_{xyy}(x, y) + v^3 f_{yyy}(x, y) \right] + \]

Third partial derivatives

+ \ldots \text{ (Higher order terms)}

First order approx

\[ f(x+u, y+v) \approx f(x, y) + uf_x(x, y) + vf_y(x, y) \]
Harris Corner Derivation

\[ \sum [I(x + u, y + v) - I(x, y)]^2 \]

\[ \approx \sum [I(x, y) + uI_x + vI_y - I(x, y)]^2 \quad \text{First order approx} \]

\[ = \sum u^2I_x^2 + 2uvI_xI_y + v^2I_y^2 \]

\[ = \sum \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad \text{Rewrite as matrix equation} \]

\[ = \begin{bmatrix} u & v \end{bmatrix} \left( \sum \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \]
Harris Detector formulation

This measure of change can be approximated by:

\[ E(u, v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \]

where \( M \) is a \( 2 \times 2 \) matrix computed from image derivatives:

\[ M = \sum_{x, y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \]

Gradient with respect to \( x \), times gradient with respect to \( y \)

Sum over image region – area we are checking for corner

Source: R. Szeliski
Harris Detector formulation

where $M$ is a $2 \times 2$ matrix computed from image derivatives:

$$M = \sum_{x, y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Sum over image region – area we are checking for corner

Source: R. Szeliski
Interpreting the eigenvalues of $M$

Classification of image points using eigenvalues of $M$:

- **“Corner”**: $\lambda_1$ and $\lambda_2$ are large, $\lambda_1 \sim \lambda_2$; $E$ increases in all directions.
- **“Edge”**: $\lambda_1 \gg \lambda_2$.
- **“Flat”** region: $\lambda_1$ and $\lambda_2$ are small; $E$ is almost constant in all directions.

Source: R. Szeliski
Corner response function

\[ R = \det(M) - \alpha \text{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2 \]

\( \alpha: \) constant (0.04 to 0.06)

Source: R. Szeliski
Algorithm steps:

- Compute $M$ matrix within all image windows to get their $R$ scores
- Find points with large corner response ($R > \text{threshold}$)
- Take the points of local maxima of $R$
Harris Detector: Properties

* Rotation invariance

Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner response $R$ is invariant to image rotation
Harris Detector: Properties

* Not invariant to image scale

All points will be classified as edges

Corner!
Noise, Thresholding, Incompleteness

Kalkan et al., Int. Conf. on Computer Vision Theory and Applications, 2007.
Another problem: Localization

Kalkan, 2008; Kalkan et al., 2007.
A corner is where lines intersect.
Since we know the edges and their orientation, we can compute whether the lines in a window are intersecting at the center.

\[ ic(p_c) = \int [c_{i1D}(p)]^2 \left[ 1 - d(l^p, p_c)/d(p, p_c) \right] dp, \]

Kalkan, 2008; Kalkan et al., 2007.
Figure 4.2: Illustration of the maximum IC for a few examples.
Kalkan, 2008; Kalkan et al., 2007.
Kalkan et al., Int. Conf. on Computer Vision Theory and Applications, 2007.
<table>
<thead>
<tr>
<th>Original Detection</th>
<th>Improved Positioning</th>
<th>Semantic Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image 1]</td>
<td>![Image 2]</td>
<td>![Image 3]</td>
</tr>
<tr>
<td>![Image 7]</td>
<td>![Image 8]</td>
<td>![Image 9]</td>
</tr>
</tbody>
</table>

Kalkan et al., Int. Conf. on Computer Vision Theory and Applications, 2007.
Kalkan et al., Int. Conf. on Computer Vision Theory and Applications, 2007.
Problems with Corner Detection

- Localization
- Representation
- Viewpoint
- Scale
Texture
Texture

* What is texture?
  * No unique definition.

* Certain aspects:
  * Repetition
  * Sometimes random
  * Sometimes involving “edges”
  * …
Why study texture?

* Because the world is full of them.
Julesz

* Textons: analyze the texture in terms of statistical relationships between fundamental texture elements, called “textons”.
Textures are made up of repeated local patterns, so:

- Find the patterns
  - Use filters that look like patterns (spots, bars, raw patches...)
  - Consider magnitude of response

- Describe their statistics within each local window
  - Mean, standard deviation
  - Histogram
  - Histogram of “prototypical” feature occurrences
Texture representation: example

original image

derivative filter responses, squared

<table>
<thead>
<tr>
<th>Window #</th>
<th>( \text{mean } d/dx ) value</th>
<th>( \text{mean } d/dy ) value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
</tbody>
</table>
Texture representation: example

- **original image**
- **derivative filter responses, squared**
- **statistics to summarize patterns in small windows**

<table>
<thead>
<tr>
<th>Window</th>
<th>mean d/dx value</th>
<th>mean d/dy value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
<td>7</td>
</tr>
</tbody>
</table>

Slide: Trevor Darrell
Texture representation: example

original image

derivative filter responses, squared

<table>
<thead>
<tr>
<th>Window</th>
<th>mean $d/dx$ value</th>
<th>mean $d/dy$ value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>Win. #9</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

statistics to summarize patterns in small windows

Slide: Trevor Darrell
Texture representation: example

Statistics to summarize patterns in small windows

Slide: Trevor Darrell
Texture representation: example

Windows with primarily horizontal edges

Windows with small gradient in both directions

Windows with primarily vertical edges

Both

<table>
<thead>
<tr>
<th>Dimension 1 (mean d/dx value)</th>
<th>Dimension 2 (mean d/dy value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
</tr>
<tr>
<td>Win. #9</td>
<td>20</td>
</tr>
</tbody>
</table>

statistics to summarize patterns in small windows

Slide: Trevor Darrell
Texture representation: example

original image

derivative filter responses, squared

visualization of the assignment to texture "types"

Slide: Trevor Darrell
Texture representation: example

Dimension 1 (mean d/dx value)

Dimension 2 (mean d/dy value)

Far: dissimilar textures

Close: similar textures

<table>
<thead>
<tr>
<th>Window</th>
<th>mean d/dx value</th>
<th>mean d/dy value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win. #1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Win. #2</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>Win. #9</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

statistics to summarize patterns in small windows

Slide: Trevor Darrell
Problem: Scale

- We’re assuming we know the relevant window size for which we collect these statistics.

Possible to perform scale selection by looking for window scale where texture description not changing.

Slide: Trevor Darrell
Texture Analysis
Using Oriented Filter Banks

Malik J, Perona P. Preattentive texture discrimination with early vision mechanisms.
J OPT SOC AM A 7: (5) 923-932 MAY 1990
Texture Analysis
Using Oriented Filter Banks

Slide: Trevor Darrell
Texture Analysis
Using Oriented Filter Banks

Modelling I – Learning the Texton Dictionary

http://www.robots.ox.ac.uk/~vgg/research/texclass/with.html
Problems involving texture

Texture Segmentation


Texture Classification

Shape from Texture

Texture Synthesis
Figure 11.14. The values of pixels at coarse scales in a pyramid are a function of the values in the finer scale layers. We associate a parent structure with each pixel, which consists of the values of pixels at coarse scales which are used to predict our pixel’s value in the Laplacian pyramid, as indicated in this schematic drawing. This parent structure contains information about the structure of the image around our pixel for a variety of differently sized neighbourhoods.
Laplacian pyramid

\[ G_1 x_1 = x_2 \]

\[ F_1 G_1 x_1 \]

\[ (I - F_1 G_1) x_1 \]

\[ (I - F_2 G_2) x_2 \]

\[ (I - F_3 G_3) x_3 \]
Laplacian pyramid

Freeman
Figure 8: The distribution from which pixels in the synthesis pyramid are sampled is conditioned on the “parent” structure of those pixels. Each element of the parent structure contains a vector of the feature measurements at that location and scale.

Figure 9: An input texture is decomposed to form an analysis pyramid, from which a new synthesis pyramid is sampled, conditioned on local features within the pyramids. A filter bank of local texture measures, based on psychophysical models, are used as features.

De Bonet,
1997.
Markov Chains

Markov Chain

- a sequence of random variables $x_1, x_2, \ldots, x_n$

- $x_t$ is the state of the model at time $t$

  $\begin{array}{c}
  x_1 \\
  \rightarrow \\
  x_2 \\
  \rightarrow \\
  x_3 \\
  \rightarrow \\
  x_4 \\
  \rightarrow \\
  x_5
  \end{array}$

- Markov assumption: each state is dependent only on the previous one
  - dependency given by a conditional probability:

  $$p(x_t | x_{t-1})$$
Markov Chain Example: Text

“\textquote{A dog is a man’s best friend. It’s a dog eat dog world out there.}”

\[
p(x_t | x_{t-1})
\]

\[
X_{t-1}
\]

\[
X_t
\]

Source: S. Seitz
Text synthesis

Create plausible looking poetry, love letters, term papers, etc.

Most basic algorithm

1. Build probability histogram
   - find all blocks of $N$ consecutive words/letters in training documents
   - compute probability of occurrence $p(x_t|x_{t-1}, \ldots, x_{t-(n-1)})$

2. Given words $x_1, x_2, \ldots, x_{k-1}$
   - compute $x_k$ by sampling from $p(x_t|x_{t-1}, \ldots, x_{t-(n-1)})$

WE NEED TO EAT CAKE
Text synthesis

Results:

• “As I've commented before, really relating to someone involves standing next to impossible.”

• "One morning I shot an elephant in my arms and kissed him."

• "I spent an interesting evening recently with a grain of salt"

Synthesizing Computer Vision text

* What do we get if we extract the probabilities from the F&P chapter on Linear Filters, and then synthesize new statements?

This means we cannot obtain a separate copy of the best studied regions in the sum.

All this activity will result in the primate visual system.

The response is also Gaussian, and hence isn’t bandlimited.

Instead, we need to know only its response to any data vector, we need to apply a low pass filter that strongly reduces the content of the Fourier transform of a very large standard deviation.

It is clear how this integral exist (it is sufficient for all pixels within a $2k + 1 \times 2k + 1 \times 2k + 1 \times 2k + 1$ — required for the images separately.
A Markov random field (MRF)

- generalization of Markov chains to two or more dimensions.

First-order MRF:

- probability that pixel X takes a certain value given the values of neighbors A, B, C, and D:

\[ P(X|A, B, C, D) \]
Texture Synthesis [Efros & Leung, ICCV 99]

* Can apply 2D version of text synthesis
Texture Synthesis by Non-parametric Sampling

Alexei A. Efros and Thomas K. Leung
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720-1776, U.S.A.
{efros,leungt}@cs.berkeley.edu

• Model the local conditional dependency of pixels using Markov Random Field.
Synthesis results

white bread

brick wall

Slide from Alyosha Efros, ICCV 1999
Synthesis results

Slide from Alyosha Efros, ICCV 1999
Hole Filling

Slide from Alyosha Efros, ICCV 1999
Input texture

Random placement of blocks

Neighboring blocks constrained by overlap

Minimal error boundary cut
Minimal error boundary

overlapping blocks

vertical boundary

overlap error

min. error boundary

Slide from Alyosha Efros
More on texture


Chapter 2.1

Texture Analysis

Mihran Tuceryan

Department of Computer and Information Science,
Indiana University -Purdue University at Indianapolis,
723 W. Michigan St. Indianapolis, IN 46202-5132

and

Anil K. Jain

Computer Science Department, Michigan State University
East Lansing, MI 48824-1027 USA
Internet: jain@cps.msu.edu
Problems with Texture

* Representation
  * Scale
  * View-point
* Matching
Segmentation

Why study segmentation?
Segmentation as Clustering

* Merging Clustering

**Algorithm 15.3:** Agglomerative clustering, or clustering by merging

- Make each point a separate cluster
- Until the clustering is satisfactory
  - Merge the two clusters with the smallest inter-cluster distance
- end

* Divisive Clustering

**Algorithm 15.4:** Divisive clustering, or clustering by splitting

- Construct a single cluster containing all points
- Until the clustering is satisfactory
  - Split the cluster that yields the two components with the largest inter-cluster distance
- end
### Algorithm 15.5: Clustering by K-Means

Choose \( k \) data points to act as cluster centers
Until the cluster centers are unchanged
  Allocate each data point to cluster whose center is nearest
  Now ensure that every cluster has at least
  one data point; possible techniques for doing this include
  supplying empty clusters with a point chosen at random from
  points far from their cluster center.
Replace the cluster centers with the mean of the elements in their clusters.
end
K-means clustering using intensity alone and color alone

Slide: A. Torralba
K-means using color alone, 11 segments
Including spatial relationships

Augment data to be clustered with spatial coordinates.

\[
\begin{pmatrix}
Y \\
u \\
v \\
x \\
y
\end{pmatrix}
\]

\(z = \begin{pmatrix}
Y \\
u \\
v \\
x \\
y
\end{pmatrix}\)

\{color coordinates\}

\{spatial coordinates\}
Mean Shift Algorithm

1. Choose a search window size.
2. Choose the initial location of the search window.
3. Compute the mean location (centroid of the data) in the search window.
4. Center the search window at the mean location computed in Step 3.
5. Repeat Steps 3 and 4 until convergence.

The mean shift algorithm seeks the “mode” or point of highest density of a data distribution:

Slide: A. Torralba
Mean Shift Segmentation

1. Convert the image into tokens (via color, gradients, texture measures etc).
2. Choose initial search window locations uniformly in the data.
3. Compute the mean shift window location for each initial position.
4. Merge windows that end up on the same “peak” or mode.
5. The data these merged windows traversed are clustered together.

Slide: A. Torralba
Mean Shift Segmentation

Segmented "landscape 1"

Segmented "landscape 2"

http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

Slide: A. Torralba
Mean Shift color&spatial Segmentation Results:

http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html
Mean Shift color & spatial Segmentation Results:

Original "fagaras"

Original "building"

Segmented

Segmented

Slide: A. Torralba
Minimum Cut and Clustering

Slide: A. Torralba

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003
Results on color segmentation


Slide: A. Torralba
Contains a large dataset of images with human "ground truth" labeling.
Do we need recognition to take the next step in performance?
Aim

* Given an image and object category, to segment the object

Segmentation should (ideally) be
- shaped like the object e.g. cow-like
- obtained efficiently in an unsupervised manner
- able to handle self-occlusion
Feature-detector view
Object-Specific Figure-Ground Segregation

Some segmentation/detection results

Slide: A. Torralba

Yu and Shi, 2002
Implicit Shape Model - Liebe and Schiele, 2003

Interest Points

Matched Codebook Entries

Probabilistic Voting

Backprojected Hypotheses

Backprojection of Maxima

Voting Space (continuous)

Slide: A. Torralba
Problems with Segmentation

* Determining similarity between pixels.
* Determining “k” or a threshold.
* Representing them.
  * Implicit vs. explicit.
* Matching.
Perceptual Grouping
Not grouped

Proximity

Similarity

Similarity

Common Fate

Common Region
Parallelism
Symmetry
Continuity
Closure
Familiarity

Slide: A. Torralba
Familiarity
Influences of grouping

Grouping influences other perceptual mechanisms such as lightness perception.

Slide: A. Torralba

Contour Integration by the Human Visual System: Evidence for a Local "Association Field"

DAVID J. FIELD,* ANTHONY HAYES,† ROBERT F. HESS†

Received 2 March 1992; in revised form 9 July 1992
Contour Integration by the Human Visual System: Evidence for a Local “Association Field”

DAVID J. FIELD,* ANTHONY HAYES,† ROBERT F. HESS†

Received 2 March 1992; in revised form 9 July 1992
Perceptual Grouping


Elder, Goldberg, 2002.
What did I skip?

- Popular descriptors like:
  - SIFT
  - SURF
  - MSER
  - ...
- Contours/Boundaries
I will supply material for:

- Edges
- Corners/Junctions
- Texture
- Segmentation