Chapter 9. Computer Vision


Lecture Notes on Artificial Intelligence, Spring 2012

Summarized by Heo, Min-Oh and Lee, Sangwoo

Biointelligence Laboratory
School of Computer Science and Engineering
Seoul National University

http://bi.snu.ac.kr
Overview of early computer vision

9.1 Hints from Biology
9.2 Recognizing Faces
9.3 Computer Vision of Three-Dimensional Solid Objects
   Early vision approaches
   Early computer vision for 3D solid objects

Appendix
   Computer Vision- Shepard table
   9.3.1 An Early Vision System
   9.3.2 The “Summer Vision Project”
   9.3.3 Image Filtering
   9.3.4 Processing Line Drawing
Overview of Chapter 9

- Computer Vision - Getting information through vision by computer

- Why difficult for computer?
  - The image is a two-dimensional project of a three-dimensional scene
  - the imaging process loss information.

- Human has stereo vision
  - Providing depth information
  - Sometimes cause errors (Shepherd’s diagram)
Human Vision vs Computer Vision
Chapter 9. Computer Vision

9.1 Hints from Biology
9.1 Hints from Biology

- “What the Frog’s Eye Tells the Frog’s Brain” (Lettvin et al., 1968.)
  - Frogs only has some detectors for objects which can help for survive. (e.g. food, predator)

- Neurophysiologic research about ‘Simple cells’ (Hubel et al, 1962, 1968)
  - Certain neurons respond selectively to images and parts of images of specific shapes.
Chapter 9. Computer Vision

9.2 Recognizing Faces
9.2 Recognizing Faces

- **Early works in 1960s and 1970s**
  - Trials to automatically recognize faces
  - Panoramic Research
    - Extracting set of features by human (the coordinates of center of pupils)
    - Computer calculate distance of feature like width of mouth
    - Build the database with value of features

- **Michael Kelly**
  - Automatically detect facial features in pictures and use them to identify people
  - Find the location of features such as eyes, nose, or shoulders
  - match by nearest-neighbor method

- **Takeo Kanade**
  - Use Program coded by assembly language
  - Run on a machine with 10ms cycle CPU and 20kB main memory

- **Limitation**
  - Images should be of faces of standard scale, pose, expression, and illumination
Chapter 9. Computer Vision

9.3 Computer Vision of Three-Dimensional Solid Objects
Early Vision Approaches

- **Summer Vision Project in MIT**
  - Trial to develop a visual system

- **Two areas on early computer vision for 3D solid objects**
  - Low-level vision: constructing a representation of the image (image filtering)
  - Analyzing the line drawing as an assemblage of separate objects
Early computer vision for 3D solid objects

- **Image filtering**
  - Extracting features by image filtering using averaging windows
  - **Examples**
    - **Sobel operator**: Finding large gradient points on vertical and horizontal directions
    - **Marr-Hildreth edge detector**
    - Marr argued that “brain use edge detector as operator in produce “primal sketch”

Original image

* LoG filtering window

LoG-processed version

Passing Sobel operator

Final result

© 2012, SNU CSE Biointelligence Lab., http://bi.snu.ac.kr
Early computer vision for 3D solid objects

- Processing line drawing
  - Methods for analyzing line drawings to find objects in images
- Huffman-Clowes labeling
  - Assigning labels to the lines by the ways in which planes could come together of the object
- Impossible object case
  - Labels might be locally consistent, but globally inconsistent

© 2012, SNU CSE Biointelligence Lab., http://bi.snu.ac.kr
Chapter 9. Computer Vision

Appendix
Computer Vision - Shepard table

Figure 9.1: Two tables (Illustration courtesy of Michael Bach.)

9.3.1 An Early Vision System

- Roberts’s System
  - Identify objects in black-and-white photographs and determine their orientation and position in space.
  - Produce a representation of a line drawing.
- Roberts Cross
  - Detecting abrupt brightness changes in an image

Figure 9.2: Detecting changes in intensity.
Figure 9.3: Producing the final line drawing.
9.3.2 The “Summer Vision Project”

- Trial to develop a visual system in MIT
  - Goal
    - Building a vocabulary of known objects
    - Analyzing a picture from a “videsector” (scanner)
    - Naming objects by matching

- Two areas on early computer vision
  - Low-level vision: constructing a representation of the image (image filtering)
  - Analyzing the line drawing as an assemblage of separate objects
9.3.3 Image Filtering

- Why filtering images?
  - correct for noise
  - enhance image features.

- Averaging Window
  - Image averaging takes into account adjacent values and combine them
  - These intensity values correspond to an image whose right side is bright and whose left is dark with a sharp edge.

Figure 9.4: An array of image intensity values and an averaging window.
9.3.3 Image Filtering

- **Sobel Operator**
  - Operator use two filtering windows
  - sensitive to large gradients in the vertical direction
  - sensitive to large gradients in the horizontal direction
  - The image intensity at each point is multiplied by the number in the corresponding cell of the filtering window before adding all of the numbers.

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

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

Figure 9.5: Sobel's vertical (left) and horizontal (right) filters.

Figure 9.6: Finding abrupt changes in image brightness with the Sobel Operator.
9.3.3 Image Filtering

- **Marr-Hildreth edge detector**
  - Laplacian of Gaussian (LoG)
    - Mexican hat
    - Sombrero function
  - LoG-processing
    - Giving whitish bands surrounding darker parts
  - Finding the transitions from light to dark (and vice versa)
  - Marr argued that “brain use edge detector as operator in produce “primal sketch”

![Original image](image1.png) ![LoG-processed version](image2.png) ![Final result](image3.png)
9.3.4 Processing Line Drawing

- Methods for analyzing line drawings to find objects in images
- Guzman-Arenas’ SEE
  - Developed LISP program called SEE
  - Procedure
    - Sorting its vertices into a number of different types.
    - For each vertex, depending on its type, SEE connected adjacent planar surfaces with “links.”
  - Capable of finding bodies in rather complex scenes.
  - Could not identify blocks that had holes in them.

Figure 9.11: Links established by SEE for a sample scene.

Figure 9.12: A scene analyzed by SEE.
9.3.4 Processing Line Drawing

- Huffman’s Work (and Max Clowes)
  - Huffman-Clowes labeling
    - Assigning labels to the lines by the ways in which planes could come together of the object
  - T-nodes
    - Four types in which three plane surface can come together at a vertex
    - Line intersection types caused by one object in a scene occluding another
  - Impossible object case
    - Labels might be locally consistent, but globally inconsistent

Figure 9.13: The four different kinds of vertices that can occur in trihedral solids.

Figure 9.14: An impossible object.
9.3.4 Processing Line Drawing

- Waltz’s Work
  - Extend Huffman’s scheme to allow for line drawings of scenes with shadows and possible “cracks” between two adjoining object.
  - Propose and implement an efficient computational method for satisfying the constraint that all of the lines must be assigned only one label.

Figure 9.15: A scene with shadows analyzed by Waltz's program.