Computer and Machine Vision

Lecture Week 10
Part-2
Skeletal Models and Face Detection
Outline of Week 10

- Lab #4 Overview
- Lab #5 and #6 Extended Lab Overview
- SIFT and SURF High Level View
- Skeletal Models and Shape Recognition
- Introduction to High Level Face Detection and Recognition with OpenCV
Skeletal Models

- Easier to Recognize Shape when Thinned Down to Skeleton
  - Thinning is a process of Finding Edges and reducing Width
  - Transformation

- Read E.R. Davies Supplemental Chapter 2 Materials for Image Operation Notation

- Read E.R. Davies Chapter 9 for Binary Shape Analysis Algorithm
Study **skeletal.cpp**

- Pre-processing
- Understand Process (Loop)
- Exit Conditions
OpenCV Skeletal Transform – Step 1

1. Note that We Apply to Graymap

   - Load Color JPG
     - Mat src = imread(filename, CV_LOAD_IMAGE_COLOR);

   - Convert to Graymap
     - cvtColor(src, gray, CV_BGR2GRAY);
     - Use common conversion – 0.3*Red + 0.59*Green + 0.11*Blue over All Pixels
OpenCV Skeletal Transform – Step 2

2. Convert Grayscale to Binary Image
   – Simple Threshold as Done in Lab #3
   – Consider Histogram Analysis
     - `threshold(gray, binary, 70, 255, CV_THRESH_BINARY);`
3. Loop to Erode and Dilate Image Iteratively
   – OpenCV Erode and Dilate Example and Tutorial
   – Just Like E.R. Davies Thinning (P. 244-250)
   – Specifically Davies Algorithm on P. 249
OpenCV Skeletal Transform

4. Exit Condition is Thinness
   – Specify Pixel Width for Thinning
   – Just Like E.R. Davies Thinning (P. 249)
     “finished” condition to exit do loop

Challenge Has Been the Computational Complexity of Otherwise Simple Transformation Logic
Value of Skeletal to Shape Recognition

- Skeletal Shape is Invariant to Lighting
  - Binary Threshold (No Color)
  - Within Reason for Extremely Shadowed or Washed out Images

- Not Scale or Rotationally Invariant, But Less Sensitive to Noise in Images

- Useful for Gate Analysis and Behavior
For Skeletal Ground Up - Start with Threshold Assumption

- Assume that Threshold transforms image to 1’s on background of 0’s, or PBM from PGM
- Define What Connected Means
- Assertion that Foreground is 8-connected (all nearest neighbors) and Background is 4-connected (only orthogonal neighbors)
Label Connected Regions (Skeletal Model Concepts)

- Raster Image with Definition of Connectedness for FG and BG
- Uniquely Assign ID to Each Connected Region
- Filter Regions Below a Size Threshold
- Skeleton is the Medial Line Along Limbs or Regions

Skeletal Models for Motion Tracking

Like Fiducial Markers on Robotic System, used to Track Limbs (Kinematics, inverse Kinematics for rigid bodies)

Skeletal Models to Robotics

Useful for Basic Gesture Recognition
- Requires Clear Separation of FG/BG
- Works Better with Stable Camera (Stereo)
- Motion in Near Foreground

Used for Generalized Recognition in 3D Along with Active Depth Mapping
- 20 Joints per Player, 6 people in theory
- 640x480 RGB, 8-bit
- SDK Popularized by Willow Garage and ROS
  - http://www.willowgarage.com/pages/software/ros-platform
Distance Metric for 8-Connected Foreground

\[ d_8 = \max(|x_i - x_j|, |y_i - y_j|) \]

Computes Distance From Edge Such that Center of Object Has Highest Value

**FIGURE 9.6**
The distance function of a binary shape: the value at every pixel is the distance (in the \( d_8 \) metric) from the background.

Method to Compress Objects

Object Shapes Described by Max Distance Values At Medial Locations

Compressed Form, Can be Expanded by Propagation to Min Distance Edge When Rendered

See Page 243, Figure 9.7 in CMV, E.R. Davies
Flood Fill

- Fills in Shape Between Defined Edges
- 4-Connected vs. 8-Connected
- Target Color (Shade) and Replacement Color (Shade) are used with Starting Pixel and Algorithm Runs Recursive
- E.g. 4-Connected Flood Fill

From Wikipedia

Flood-fill (node, target-color, replacement-color):
1. If the color of node is not equal to target-color, return.
2. Set the color of node to replacement-color.
3. Perform Flood-fill (one step to the west of node, target-color, replacement-color).
   Perform Flood-fill (one step to the east of node, target-color, replacement-color).
   Perform Flood-fill (one step to the north of node, target-color, replacement-color).
   Perform Flood-fill (one step to the south of node, target-color, replacement-color).
4. Return.
Thinning (Skeletons)

- Useful for Character Recognition and Gesture Recognition
  - Eliminates Redundant Shape Information and Preserves Topological Information Only (a Form of Compression)
  - Connected Set of Medial Lines Along Limbs

Fig 9.8 Mathematical Idealized Medians

Fig 9.9 From Thinning Algorithm

Chi Function

Used To Determine Which Pixels Can Be Removed and Which Must Be Retained

- If Pixel is on Boundary, it Can Be Removed and Chi=2, Sigma != 1
- Sigma is Sum of Neighbors (Where FG=1, BG=0)
- Chi is a Complex Conditional Expression to Encode When Pixel of Interest is On An Edge and Can Be Removed
- Note Edge Conditions to Understand Chi

<table>
<thead>
<tr>
<th>Line End</th>
<th>N edge</th>
<th>E edge</th>
<th>S edge</th>
<th>W edge</th>
<th>NW edge</th>
<th>SE edge</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>1 1 1</td>
<td>0 1 1</td>
<td>0 0 0</td>
<td>1 1 0</td>
<td>1 1 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>1 1 1</td>
<td>0 1 1</td>
<td>1 1 1</td>
<td>1 1 0</td>
<td>1 1 0</td>
<td>0 1 1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 0 0</td>
<td>0 1 1</td>
<td>1 1 1</td>
<td>1 1 0</td>
<td>1 1 0</td>
<td>1 1 1</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 0 0</td>
<td>0 1 1</td>
<td>1 1 1</td>
<td>1 1 0</td>
<td>1 1 0</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

© Sam Siewert
Thinning Algorithm Chi

Consider All Cases of Proper Edges Where Pixel Can Be Thinned (Chi=2) and Sigma != 1

\[
\text{Chi} = (A_1 \neq A_3) + (A_3 \neq A_5) + (A_5 \neq A_7) + (A_7 \neq A_1) + 2 \times ((A_2 > A_1) \&\& (A_2 > A_3)) + ((A_4 > A_3) \&\& (A_4 > A_5)) + ((A_6 > A_5) \&\& (A_6 > A_7)) + ((A_8 > A_7) \&\& (A_8 > A_1))
\]

E.g. Here are Some Non-Edges to Retain

<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>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Isolated

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

Chi=4

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

Chi=8
Combined Distance and Thinning for Guided

Compute Maximum Distance and Thin – Better for Parallel Speed-up (P. 252 CMV, E.R. Davies)

Intro to Face Detection

Pattern Matching Well Known Features
- Eyes
- Nose
- Mouth

Characteristics of the Pattern
- Eye Separation, Eyes to Nose, Nose to Mouth
- If We can See all Three Features (and Detect) a Face Has Similar Components (Distance) Measure for Features
- Viewing Angle, Occlusion, Lighting Complicates
Evidence of Scanning

- Human Saccades – Fast Human Eye Movement used in Facial Scanning

- Study with Pupil Tracker

- 900 degrees / sec rate

- 200 millisecond to initiate, 20-200 millisecond to complete

http://en.wikipedia.org/wiki/Saccades
Haar Features

- Pages 506-516 in Learning OpenCV
- Transform the Image using Haar Wavelets
- Facial Features – Eyes, Nose, Mouth and Shape of Face are Simpler in the Transform and Better Invariance
- Match Transformed Test Image to Haar Transform Database
- Note Regions of Interest in Test Image
Facial Detection and Recognition Research

- Viola and Jones – Robust Real-Time Face Detection (See Presentation and Paper on BB)

- Ada Boost

- Open Research Today – Value to Security, Photo Databases, Social Networking …