Announcement

Quiz #10 is available in Blackboard.

Due date: 11:59pm EST, Wednesday, April 14th

Open book and open notes
Materials Covered in the Presentation

For a research/hands-on project
• An introduction of the background
• A brief literature review
• Methodology of your proposed method
• Experimental results if any

For a survey project
• An introduction of the background
• A discussion on the papers you reviewed
• Comparison of the methods/groups you reviewed

Please submit your final presentation slides into Blackboard before your presentation.
Estimate Camera Parameters from $P$

$P = \zeta \begin{bmatrix} f_u r_1^t + u_0 r_3^t & f_u t_X + u_0 t_z \\ f_v r_2^t + v_0 r_3^t & f_v t_Y + v_0 t_z \\ r_3^t \end{bmatrix} = \begin{bmatrix} p_1^t & p_{14} \\ p_2^t & p_{24} \\ p_3^t & p_{34} \end{bmatrix}$

$r_3 = \zeta p_3 \quad t_z = \zeta p_{34}$

How to determine $\zeta$

$\zeta$ is determined from $t_z$

$t_z$ is negative if the object is before the camera
Verifying Rectification

Before rectification

After rectification

Figure from Stack Overflow
How to Draw Epipolar Lines using Fundamental Matrix

Given \( F \) and a point on one image, we can estimate the corresponding epipolar line

\[
U_l^t F U_r = 0
\]

Epipolar constraint: \( U_l^t (FU_r) = 0 \)

Recall line function: \( l \cdot p = 0 \), where \( l \) represents the line parameters \( ax + by + c = 0 \)

\[
l_{el} = FU_r
\]

\( U_l \) is on a line determined by \( l_{el} \)
Epipolar line on the left image corresponding to \( U_r \) on the right image
Questions in Quiz 9

Model-free image registration such as rigid-body transformation and free-form transformation assumes that the two images are viewing the same scene/object with small appearance difference.

Pros:
- General to arbitrary scenes and objects
- Registration can be performed on the whole image
- No need of training data

Cons:
- Suffer from large shape/appearance changes caused by large variations in view, identity, illumination, occlusion, etc.
- Susceptible to noise
Questions in Quiz 9

Model-based method assumes that the target is an instantiation of a known class.

Pros:
- May handle large shape/appearance variations depending on the models
- Robust to noise

Cons:
- Need sufficient training data
- Usually needs a good initialization
- The target should follow the same distribution as the training data
- Registration can be only performed on the specific target
Today

Image segmentation
Hierarchy of Computer Vision Problems

- **Sensing**
  - Computational model of camera
  - Radiometry
  - Camera calibration, etc.

- **Low-level Information**
  - Single image: filters, edges, features, etc.
  - Multiple images: stereo vision, motion analysis, etc.

- **Mid-level Information**
  - Segmentation and grouping
  - Object tracking, image registration

- **High-level Information/Understanding**
  - Object recognition
  - Scene understanding
  - Image interpretation, etc.
Image Segmentation

A process that partitions $R$ into subregions $R_1, R_2, \ldots, R_n$
Image Segmentation – Applications

Object localization

Object recognition

Specifically important for medical imaging
**Brief Review of Connectivity**

- Path from \( p \) to \( q \): a sequence of distinct pixels with coordinates \((x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\) starting point \( p \) ending point \( q \)

- \( p \) and \( q \) are *connected*: if there is a path from \( p \) to \( q \) in \( S \)

- *Connected component*: all the pixels in \( S \) connected to \( p \)

- *Connected set*: \( S \) has only one connected component

- \( R \) is a region if \( R \) is a connected set

- \( R_i \) and \( R_j \) are adjacent if \( R_i \cup R_j \) is a connected set
Two categories based on intensity properties:

- **Discontinuity** – edge-based algorithms
- **Similarity** – region-based algorithms
Edge-based and Region-based Segmentation

Figure 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.

Figure from “Digital Image Processing”, 3rd edition, Gonzalez and Woods
Brief Review on Simple Edge Detectors

- **First-order derivative**
  - E.g., Roberts (2x2), Prewitt (3x3), Sobel (3x3, smooth + difference)
  - Thicker edge
  - One operator for one edge direction

- **Second-order derivative**
  - Laplacian (3x3)
  - Double edge
  - Zero-crossing

- **Common issues:**
  - Sensitive to image noise
  - Cannot deal with the scale change of the image
Advanced Edge Detection Techniques

- Deal with image noise
- Exploit the properties of image
  Work much better for real images

Advanced edge detectors:
- Laplacian of Gaussian (LoG) \( \nabla^2 G(x, y) = \left[ \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2+y^2}{2\sigma^2}} \)
- Difference of Gaussian (DoG)
  \[
  DOG(x, y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}, \quad \sigma_1 > \sigma_2
  \]
- Canny
Marr-Hildreth Detector (LoG)

**Observations:**
- Intensity changes are dependent on the image scale
- A sudden intensity change (step) causes a peak/trough in the 1\textsuperscript{st} order derivative and a zero-crossing in the 2\textsuperscript{nd} order derivative
- The 2\textsuperscript{nd} order derivative is especially sensitive to noise

Smooth the image using a Gaussian filter first before applying the Laplacian

\[ G(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}} \]

\[ g(x, y) = \nabla^2 [G(x, y) \otimes f(x, y)] = \nabla^2 G(x, y) \otimes f(x, y) \]

- Varying \( \sigma \) values for scale changes
- Rotation invariant in edge detection
LoG Filtering

\[ g(x, y) = \nabla^2 G(x, y) \otimes f(x, y) \]
\[ = \nabla^2 [G(x, y) \otimes f(x, y)] \]

1. Filter the input image with an \( nxn \) Gaussian filter.

2. Compute the Laplacian of the intermediate image resulting from Step 1.

3. Find the zero-crossings of the image from Step 2.
   - opposite signs of the neighbors
   - the difference should be significant

**Note:**
- Window size \( n \geq 6 \sigma \) and \( n \) is an odd number
An Example – Edges are 1 Pixel Thick

Original | LoG filtering | LoG filtering with $T=0$

Zero-crossing with $T=0$ | Zero-crossing with $T=4\% \text{max}$

Figure from “Digital Image Processing”, 3rd edition, Gonzalez and Woods
Approximate LoG by DoG

LoG needs multiple filters for scale variations

Difference of Gaussian (DoG)

\[
DOG(x, y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}, \quad \sigma_1 > \sigma_2
\]

Figure 10.23
(a) Negatives of the LoG (solid) and DoG (dotted) profiles using a standard deviation ratio of 1.75:1.
(b) Profiles obtained using a ratio of 1.6:1.

Figure from “Digital Image Processing”, 3rd edition, Gonzalez and Woods
Canny Edge Detector

The general goal of edge detection:

1. Low error rate: all edges should be found and there should be no false response

2. Edge points should be well localized: the edges located must be as close as possible to the true edges

3. Single edge point response: the detector should return only one point for each true edge point

Canny edge detector: expressing these three criteria mathematically and then find optimal solutions
Canny Edge Detector

Gaussian smoothing + 1\textsuperscript{st} order derivative

\[ f_s(x, y) = G(x, y) \otimes f(x, y) \]

\[ g_x = \frac{\partial f_s}{\partial x} \quad g_y = \frac{\partial f_s}{\partial y} \]

\[ M(x, y) = \sqrt{g_x^2 + g_y^2} \quad \alpha(x, y) = \tan^{-1}\left(\frac{g_y}{g_x}\right) \]

Thick edge \quad Nonmaxima suppression \quad Single edge
Quantize the Edge Direction

Figure 10.24
(a) Two possible orientations of a horizontal edge (in gray) in a $3 \times 3$ neighborhood.
(b) Range of values (in gray) of $\alpha$, the direction angle of the edge normal, for a horizontal edge. (c) The angle ranges of the edge normals for the four types of edge directions in a $3 \times 3$ neighborhood. Each edge direction has two ranges, shown in corresponding shades of gray.

Figure from “Digital Image Processing”, 3rd edition, Gonzalez and Woods
Canny Detector -- Algorithm

1. Smooth the input image with a Gaussian filter

2. Compute the gradient magnitude and angle images

3. Apply nonmaximum suppression on the gradient magnitude image
   1. At \((x, y)\), find the quantized edge normal \(d_k\)
   2. If the value \(M(x, y)\) is less than at least one of its two neighbors along \(d_k\), let \(g_N(x, y) = 0\); otherwise \(g_N(x, y) = M(x, y)\)

4. Reduce false edge: double thresholding and connectivity analysis to detect and link edges
   1. High-threshold \(\rightarrow\) strong edge pixels \(\rightarrow\) valid edge pixels
   2. Low-threshold \(\rightarrow\) weak edge pixels \(\rightarrow\) valid only when connected to strong edge pixels
FIGURE 10.25
(a) Original image of size $834 \times 1114$ pixels, with intensity values scaled to the range $[0, 1]$.
(b) Thresholded gradient of smoothed image.
(c) Image obtained using the Marr-Hildreth algorithm.
(d) Image obtained using the Canny algorithm. Note the significant improvement of the Canny image compared to the other two.

Figure from “Digital Image Processing”, 3rd edition, Gonzalez and Woods
Figure 10.26
(a) Original head CT image of size $512 \times 512$ pixels, with intensity values scaled to the range $[0, 1]$. 
(b) Thresholded gradient of smoothed image.
(c) Image obtained using the Marr-Hildreth algorithm.
(d) Image obtained using the Canny algorithm. 
(Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)
Edge Linking and Boundary Detection

All edge detection algorithms can only detect fragments of boundaries, due to image noise, non-uniform illuminations, or other effects.

Edge linking: link edges into longer meaningful edges or a full region boundaries.

Three classic methods:
- Local processing
- Regional Processing
- Hough transform
Edge Linking – Local Processing

Link the edge points with similar properties:
  • Strength
  • Direction

Two edge pixels are linked if

\[ |M(s, t) - M(x, y)| \leq E \]
\[ |\alpha(s, t) - \alpha(x, y)| \leq A \]
Global Processing

A basic idea: Given $n$ points

1. Find $n(n-1)/2$ lines between each pair of points

2. Find all subset of points that are close to particular lines. This needs $n(n-1)n/2$ comparisons

This is computationally expensive!

Hough transform
Hough Transform

A line in x-y plane is a point in the parameter plane. A point in x-y plane is a line in the parameter plane.

X-Y plane  Parameter plane

Problem of slope-intercept form: the slope $a$ approaches infinity for vertical lines
Hough Transform

A line in x-y plane is a point in the parameter plane.
A point in x-y plane is a sinusoidal in the parameter plane (polar space).

\[ x \cos \theta + y \sin \theta = \rho \]

**Figure 10.32** (a) \((\rho, \theta)\) parameterization of line in the xy-plane. (b) Sinusoidal curves in the \(\rho \theta\)-plane; the point of intersection \((\rho', \theta')\) corresponds to the line passing through points \((x_i, y_i)\) and \((x_f, y_f)\) in the xy-plane. (c) Division of the \(\rho \theta\)-plane into accumulator cells.

Figure from “Digital Image Processing”, 3rd edition, Gonzalez and Woods
A Toy Example

$$x \cos \theta + y \sin \theta = \rho$$

Figure from “Digital Image Processing”, 3rd edition, Gonzalez and Woods
**Hough Transform – Real Example**

Find the runway

---

**FIGURE 10.34** (a) A 502 × 564 aerial image of an airport. (b) Edge image obtained using Canny’s algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes. (e) Lines superimposed on the original image.

Figure from “Digital Image Processing”, 3rd edition, Gonzalez and Woods