Calibrated Stereo (Part 1)

Computer Vision I
CSE 252A
Lecture 7
Announcements

• Assignment 1 is due Oct 20, 11:59 PM
• Assignment 2 will be released Oct 20
  – Due Nov 3, 11:59 PM
• Reading:
  – Szeliski
    • Sections 12.1.1 and 11.3.1
Why Do We Have Two Eyes?

1. Redundancy – If we lose one, we’re not blind
2. Larger field of view
3. Ability to recover depth for some points
Why Do We Have Two Eyes?

Depth information is lost in image formation.

Binocular (stereo) vision enables depth estimation.
Stereo Vision

Holmes Stereoscope
An Application: Mobile Robot Navigation

The Stanford Cart, H. Moravec, 1979

Mobi, Stanford, 1987

INRIA Mobile Robot 1990
Mars Exploratory Rovers: Spirit and Opportunity, 2004
Curiosity Rover (2012)

- Navigation cameras (Navcams) B&W, 45° field of view
- Hazard avoidance cameras (hazcams), 4 pairs, 120° field of view
Boston Dynamics

Stereo + Lidar
Commercial Stereo Heads
Binocular Stereopsis: Mars
Given two images of a scene where relative locations of cameras are known, estimate depth of all common scene points.

Two images of Mars (Viking Lander)
Matching complexity (naïve)

For each point in left mage, there are $O(n^2)$ possible matching points in right image.

Input: two images that are $n \times n$ pixels

For a given point in the left image, where do we look in the right image?

With $n^2$ pixels in left image, complexity of matching is $O(n^4)$

Can we do better?
• **Baseline**: line connecting camera centers (of projection) C and C’
• **Epipoles (e, e’)**: Two intersection points of baseline with image planes
• **Epipolar Plane**: Any plane that contains the baseline
• **Epipolar Lines (l, l’)**: Pair of lines from intersection of an epipolar plane with the two image planes
Family of Epipolar Planes

- Epipolar Plane: Any plane that contains the baseline
- The set of epipolar planes is a family of all planes passing through the baseline and can be parameterized by the angle about baseline
Epipolar matching

- Potential matches for $p$ have to lie on the corresponding epipolar line $l'$
- Epipolar line $l'$ passes through epipole $e'$, the intersection of the baseline with the image plane
- Potential matches for $p'$ have to lie on the corresponding epipolar line $l$
Epipolar matching complexity

Using epipolar matching, complexity is reduced from $O(n^4)$ to $O(n^3)$. Why?
- There are $n^2$ points in the left image
- For each point in the left image, all candidate matches are on an epipolar line in the right image, and the length of the epipolar line is $O(n)$
- Therefore, match complexity is $O(n^2 \times n) = O(n^3)$
Stereo Vision Outline

• Offline
  – Calibration of stereo cameras

• Online
  1. Acquire stereo images
  2. Epipolar rectify stereo images
  3. Establish correspondence
  4. Estimate depth
Calibration of stereo cameras

1. From images of known calibration fixture, determine intrinsic parameters $K_1$, $K_2$ and extrinsic relation of two cameras $R_1$, $t_1$ and $R_2$, $t_2$

2. Compute the relative rotation $R$ and translation $t$ of the two cameras from $R_1$, $t_1$ and $R_2$, $t_2$

3. Compute the essential matrix $E$
Camera calibration

• Given $n$ points $P_1, \ldots, P_n$ with known 3D position and their pixel coordinates $x_1, \ldots, x_n$, estimate intrinsic $K$ (and lens distortion parameters) and extrinsic camera parameters $R, t$

• See textbook for details

• Camera Calibration Toolbox for Matlab (Bouguet)
  
  http://www.vision.caltech.edu/bouguetj/calib_doc/
Compute the rotation and translation of the second camera relative to the first one

\[
x_1 = K_1[I | 0] \begin{bmatrix} R_1 & t_1 \\ 0^T & 1 \end{bmatrix} X
\]
\[
x_1 = K_1[I | 0]X_{\text{cam},1}
\]

where \( X_{\text{cam},1} = \begin{bmatrix} R_1 & t_1 \\ 0^T & 1 \end{bmatrix} X \)

\[
\begin{bmatrix} R_1 & t_1 \\ 0^T & 1 \end{bmatrix}^{-1} X_{\text{cam},1} = X
\]

\[
x_2 = K_2[I | 0] \begin{bmatrix} R_2 & t_2 \\ 0^T & 1 \end{bmatrix} X
\]
\[
x_2 = K_2[I | 0] \begin{bmatrix} R_2 & t_2 \\ 0^T & 1 \end{bmatrix} \left[ \begin{bmatrix} R_1 & t_1 \\ 0^T & 1 \end{bmatrix}^{-1} \right] X_{\text{cam},1}
\]
\[
x_2 = K_2[I | 0] \begin{bmatrix} R_2 & t_2 \\ 0^T & 1 \end{bmatrix} \left[ \begin{bmatrix} R_1^T & -R_1^T t_1 \\ 0^T & 1 \end{bmatrix} \right] X_{\text{cam},1}
\]
\[
x_2 = K_2[I | 0] \begin{bmatrix} R_2 R_1^T & t_2 - R_2 R_1^T t_1 \\ 0^T & 1 \end{bmatrix} X_{\text{cam},1}
\]

where \( R = R_2 R_1^T \) and \( t = t_2 - R_2 R_1^T t_1 \)
Image points

• Image points in pixel coordinates

\[ x = K[I \mid 0] \begin{bmatrix} R & t \\ 0^\top & 1 \end{bmatrix} X \]

\[ x = K[R \mid t]X \]

• Image points in normalized coordinates

\[ x = K[R \mid t]X \]

\[ K^{-1}x = [R \mid t]X \]

\[ \hat{x} = [R \mid t]X \text{ where } \hat{x} = K^{-1}x \]
Image points in normalized coordinates

\[ \mathbf{x} = K [I | 0] \begin{bmatrix} R & \mathbf{t} \\ 0^T & 1 \end{bmatrix} \mathbf{X} \]

\[ K^{-1} \mathbf{x} = [I | 0] \begin{bmatrix} R & \mathbf{t} \\ 0^T & 1 \end{bmatrix} \mathbf{X} \]

\[ \hat{\mathbf{x}} = [I | 0] \begin{bmatrix} R & \mathbf{t} \\ 0^T & 1 \end{bmatrix} \mathbf{X} \text{ where } \hat{\mathbf{x}} = K^{-1} \mathbf{x} \]

\[ \hat{\mathbf{x}} = [I | 0] \mathbf{X}_{\text{cam}} \]

\[ \hat{\mathbf{x}} = [I | 0] \begin{bmatrix} \tilde{\mathbf{X}}_{\text{cam}} \\ 1 \end{bmatrix} \]

\[ \hat{\mathbf{x}} = \tilde{\mathbf{X}}_{\text{cam}} \text{ (up to nonzero scale)} \]
The essential matrix

3D point $\tilde{X}' = \lambda' \hat{x}'$ in the second camera coordinate frame.
3D point $\tilde{X} = \lambda \hat{x}$ in the first camera coordinate frame.
Map $\tilde{X}$ from first camera coordinate frame to second camera coordinate frame.

\[
\tilde{X}' = R \tilde{X} + t
\]
\[
\tilde{X}' = R(\lambda \hat{x}) + t
\]
\[
\tilde{X}' = \lambda R \hat{x} + t
\]
The essential matrix

3D point $\tilde{X}' = \lambda' \hat{x}'$ in the second camera coordinate frame.
3D point $\tilde{X} = \lambda \hat{x}$ in the first camera coordinate frame.
Map $\tilde{X}$ from first camera coordinate frame to second camera coordinate frame.

$$\tilde{X}' = \mathbf{R} \tilde{X} + t$$
$$\tilde{X}' = \mathbf{R} (\lambda \hat{x}) + t$$
$$\tilde{X}' = \lambda \mathbf{R} \hat{x} + t$$

$$\lambda \mathbf{R} \hat{x} + t = \lambda' \hat{x}'$$
$$[t]_\times (\lambda \mathbf{R} \hat{x} + t) = [t]_\times (\lambda' \hat{x}')$$
$$\lambda [t]_\times \mathbf{R} \hat{x} = \lambda' [t]_\times \hat{x}'$$
$$\hat{x}'^\top (\lambda [t]_\times \mathbf{R} \hat{x}) = \hat{x}'^\top (\lambda' [t]_\times \hat{x}')$$
$$\lambda \hat{x}'^\top [t]_\times \mathbf{R} \hat{x} = 0$$
$$\hat{x}'^\top [t]_\times \mathbf{R} \hat{x} = 0$$

The epipolar constraint $\hat{x}'^\top \mathbf{E} \hat{x} = 0$ where $\mathbf{E} = [t]_\times \mathbf{R}$

Essential Matrix
(Longuet-Higgins, 1981)
Cross product using a skew symmetric matrix

- The cross product $\mathbf{a} \times \mathbf{b}$ of two 3-vectors $\mathbf{a} = (a_1, a_2, a_3)^T$ and $\mathbf{b} = (b_1, b_2, b_3)^T$ can be expressed as a matrix-vector product $[\mathbf{a}]_\times \mathbf{b}$, where $[\mathbf{a}]_\times$ is the 3x3 skew symmetric matrix

$$
[\mathbf{a}]_\times = \begin{bmatrix}
0 & -a_3 & a_2 \\
 a_3 & 0 & -a_1 \\
- a_2 & a_1 & 0
\end{bmatrix}
$$

- A matrix $\mathbf{S}$ is skew symmetric if and only if $\mathbf{S} = -\mathbf{S}^T$
- The determinant of a skew symmetric matrix is 0
The essential matrix

- Maps a point (in normalized coordinates) in the first image to its corresponding epipolar line (in normalized coordinates) in the second image
  \[ \hat{\ell}' = E\hat{x} \]
- The epipolar line passes through the corresponding point in the second image
  \[ \hat{x}'^\top \hat{\ell}' = 0 \]
  \[ \hat{x}'^\top E\hat{x} = 0 \quad \text{The epipolar constraint} \]
- Every epipolar line passes through the epipole
  \[ \hat{e}'^\top \hat{\ell}' = 0 \]
The essential matrix

• Maps a point (in normalized coordinates) in the second image to its corresponding epipolar line (in normalized coordinates) in the first image

\[ \hat{l} = E^T \hat{x}' \]

\[ \hat{l}^T = \hat{x}'^T E \]

• The epipolar line passes through the corresponding point in the first image

\[ \hat{l}^T \hat{x} = 0 \]

\[ \hat{x}'^T E \hat{x} = 0 \quad \text{The epipolar constraint} \]

• Every epipolar line passes through the epipole

\[ \hat{l}^T \hat{e} = 0 \]
Example of using the essential matrix
Another example: converging cameras

courtesy of Andrew Zisserman
Another example: second camera in front of first camera

courtesy of Andrew Zisserman
Epipolar rectify stereo images
• Epipolar geometry reduces matching complexity from $O(n^4)$ to $O(n^3)$
• But matching requires comparing points across pairs of epipolar lines which may have arbitrary orientation. That can be costly to index.
• Is there a more convenient epipolar geometry
Cameras with a convenient epipolar geometry

• When two cameras have parallel optical axes and these axes are orthogonal to the baseline, the epipolar line are parallel

• When rows of the two images are parallel to the baseline, the epipolar lines are horizontal rows of the two images
Cameras with a convenient epipolar geometry

• When two cameras have parallel optical axes and these axes are orthogonal to the baseline, the epipolar line are parallel

• When rows of the two images are parallel to the baseline, the epipolar lines are horizontal rows of the two images
What if stereo geometry is not convenient?
Rectification: Given a pair of images, transform both images so that epipolar lines are image rows
Epipolar rectification

- Given a pair of images, transform both images so that epipolar lines are image rows
- Create pair of virtual cameras
  - The virtual cameras have the same camera centers as real cameras
  - Both virtual cameras have the same:
    - Camera rotation matrix $R$
    - Camera calibration matrix $K$
Epipolar rectification

- Given calibrated stereo cameras (i.e., \( K_1, R_1, t_1, K_2, R_2, t_2 \)) determine the (same) rotation matrix \( R \) and (same) calibration matrix \( K \) of the virtual cameras

- To minimize image distortion:
  - For the calibration matrix
    - For principal point, set \( x_0 \) and \( y_0 \) to average of input \( x_0 \) and \( y_0 \) values, respectively
    - Set two focal length parameters to average of all four input focal length parameters (results in square pixels)
    - Set skew to 0
  - For the rotation matrix \( R \), interpolate halfway between the two 3D rotations embodied by \( R_1 \) and \( R_2 \)
Epipolar rectification

• For the rotation matrix $\mathbf{R}$, interpolate halfway between the two 3D rotations embodied by $\mathbf{R}_1$ and $\mathbf{R}_2$
  
  – Rotating a camera does not change its center $\mathbf{C}$, but does change its translation $\mathbf{t}$

\[
\begin{align*}
\tilde{0}_{\text{cam}} &= \mathbf{R} \tilde{\mathbf{C}} + \mathbf{t} \\
-\mathbf{R} \tilde{\mathbf{C}} &= \mathbf{t} \\
\tilde{\mathbf{C}} &= -\mathbf{R}^T \mathbf{t} \\
-\mathbf{R}_{\text{virtual}} \mathbf{t}_{\text{virtual}} &= -\mathbf{R}_{\text{real}} \mathbf{t}_{\text{real}} \\
\mathbf{t}_{\text{virtual}} &= \mathbf{R}_{\text{virtual}} \mathbf{R}_{\text{real}}^T \mathbf{t}_{\text{real}} \\
\tilde{\mathbf{C}} &= -\mathbf{R}_{\text{virtual}}^T \mathbf{t}_{\text{virtual}} \\
\tilde{\mathbf{C}} &= -\mathbf{R}_{\text{real}}^T \mathbf{t}_{\text{real}}
\end{align*}
\]

\[
\begin{align*}
\mathbf{x} &= \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \mathbf{X} \\
\mathbf{x} &= \mathbf{K} \begin{bmatrix} \mathbf{R} & -\mathbf{R} \tilde{\mathbf{C}} \end{bmatrix} \mathbf{X} \\
\mathbf{x} &= \mathbf{K} \mathbf{R} \begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{C}} \end{bmatrix} \mathbf{X}
\end{align*}
\]
Rectification transformation matrices

- Transformation from image acquired by real camera to image acquired by virtual camera

\[
x_{\text{real}} = K_{\text{real}}R_{\text{real}}[I | - \tilde{C}]X
\]
\[
R_{\text{real}}^\top K_{\text{real}}^{-1} x_{\text{real}} = [I | - \tilde{C}]X
\]

\[
x_{\text{virtual}} = K_{\text{virtual}}R_{\text{virtual}}[I | - \tilde{C}]X
\]
\[
x_{\text{virtual}} = K_{\text{virtual}}R_{\text{virtual}}R_{\text{real}}^\top K_{\text{real}}^{-1} x_{\text{real}}
\]
\[
x_{\text{virtual}} = Hx_{\text{real}}, \text{ where } H = K_{\text{virtual}}R_{\text{virtual}}R_{\text{real}}^\top K_{\text{real}}^{-1}
\]

\[
\begin{bmatrix}
x_{\text{virtual}} \\
y_{\text{virtual}} \\
w_{\text{virtual}}
\end{bmatrix} =
\begin{bmatrix}
h_{11} & h_{12} & h_{13} \\
h_{21} & h_{22} & h_{23} \\
h_{31} & h_{32} & h_{33}
\end{bmatrix}
\begin{bmatrix}
x_{\text{real}} \\
y_{\text{real}} \\
w_{\text{real}}
\end{bmatrix}
\]
Rectification

Under perspective projection, the mapping from a plane to a plane is given by a 2D projective transformation (homography).
Rectification

Under perspective projection, the mapping from a plane to a plane is given by a 2D projective transformation (homography)

\[
\begin{bmatrix}
    x_L \\
    y_L \\
    w_L
\end{bmatrix}
= H_L
\begin{bmatrix}
    u_L \\
    v_L \\
    1
\end{bmatrix}
\]

Two images
Two homographies
\(H_L, H_R\)

\[
\begin{bmatrix}
    x_R \\
    y_R \\
    w_R
\end{bmatrix}
= H_R
\begin{bmatrix}
    u_R \\
    v_R \\
    1
\end{bmatrix}
\]
Image pair rectification

Apply projective transformation so that epipolar lines correspond to horizontal scanlines

\[
\begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix} = H e
\]

H should map epipole \( e \) to \((1,0,0)\), a point at infinity on the x-axis

H should minimize image distortion

Note that rectified images are usually not rectangular
Rectification

Given a pair of images, transform both images so that epipolar lines are scan lines.

Input Images
Rectification

Given a pair of images, transform both images so that epipolar lines are scan lines.

Rectified Images

epipolar lines run parallel with the x-axis and are aligned between two views (no y disparity)
Epipolar rectification, forward method

- Input: Source image: I and Rectification matrix H
- For each corner \( c_s \) of Source image in homogenous coordinates, compute \( c_t = Hc_s \)
- Compute smallest and largest x and y of \( c_t \)'s. Determine bounding box on target image. Create target image T with size of bounding box.
- For each pixel with coordinate \( p_s \) (homogenous) in the Source image, compute location in the Target image as \( p_t = Hp_s \). Copy \( I(p_s) \) to \( T(p_t) \)
Problem with Forward Method

- There’s no guarantee that every pixel in Target Image will be written to.

- If Target Image is larger than Source or Target is highly stretched, there may be missing points that appear as speckles or lines.
Epipolar rectification, backward method

• Input: Source image: I and rectification matrix H

• For each corner $c_s$ of Source image in homogenous coordinates, compute $c_t = Hc_s$

• Compute smallest and largest x and y of $c_t$’s, determine bounding box on target image, create target image T with size of bounding box.

• For each pixel coordinate $p_t$ (homogenous) in the Target, compute location in the Source as $p_s = H^{-1}p_t$.
  - If $p_s$ is within source image, copy $I(p_s)$ to $T(p_t)$
Rectification

Original

Rectified
Rectification

- Epipolar lines
Rectification

Original

Rectified
Polar Rectification

Homography-based Rectification

Polar Rectification

Alternative epipolar rectification method that minimizes pixel distortion
Polar Rectification

Epipoles are in images
(white dot on ball)

Homography-based rectification
is not possible
Features on same epipolar line
Next Lecture

• Calibrated stereo and feature matching
• Reading:
  – Szeliski
    • Sections 12.3, 12.5, 12.6