Image Rectification (Stereo)  
(New book: 7.2.1,  
old book: 11.1)  

Guido Gerig  
CS 6320 Spring 2013  

Credits: Prof. Mubarak Shah, Course notes modified from:  
http://www.cs.ucf.edu/courses/cap6411/cap5415/, Lecture 25
Example: converging cameras
Epipolar Lines in Converging Cameras

Epipolar lines all intersect at epipoles.
Stereo image rectification

In practice, it is convenient if image scanlines are the epipolar lines.
If the two cameras are aligned to be coplanar, the search is simplified to one dimension - a horizontal line parallel to the baseline between the cameras.
Intuitions on Rectification

- Let's look in 2D
- If I change the *position* of my camera, then the location of the point depends on depth
Intuitions on Rectification

- Let's look in 2D
- If I change the *position* of my camera, then the location of the point depends on depth
Intuitions on Rectification

• But if I just rotate the camera, then I don't need to know the depth

• Notice that $P'$ and $P$ still project to the same point in the rotated camera
Image Rectification

- I can rotate views to be parallel to each other and baseline
Stereo image rectification

- Image Reprojection
  - reproject image planes onto common plane parallel to line between optical centers
  - a homography (3x3 transform) applied to both input images
  - pixel motion is horizontal after this transformation
Stereo image rectification: example

Source: Alyosha Efros
Stereo image rectification: example

Source: Alyosha Efros
Stereo image rectification: example

Source: Alyosha Efros
Image Rectification
Image Rectification
Image Rectification

\[ \Pi \quad O \quad \Pi' \]

\[ e \quad P \quad O' \quad e' \]
Image Rectification
Image Rectification
Image Rectification
Image Rectification
Image Rectification
Image Rectification

- Common Image Plane
- Parallel Epipolar Lines
- Search Correspondences on scan line
- Epipoles $\rightarrow \infty$
Image Rectification
All epipolar lines are parallel in the rectified image plane.
Figure 7.15: Standard stereo setup
Essential matrix example: parallel cameras

\[
R =
\]

\[
T =
\]

\[
E = [T_x] R =
\]
Essential matrix example: parallel cameras

\[ R = I \]

\[ T = \]

\[ E = [T_x]R = \]
Essential matrix example: parallel cameras

\[
\begin{align*}
R &= I \\
T &= [-d, 0, 0]^T \\
E &= [Tx]R \\
\end{align*}
\]
Essential matrix example: parallel cameras

\[ R = I \]
\[ T = [-d, 0, 0]^T \]
\[ E = [T_x]R = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix} \]
Essential matrix example: parallel cameras

\[ R = I \]

\[ T = [-d, 0, 0]^T \]

\[ E = [T_x]R = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix} \]

\[ p'^T E p = 0 \]
Essential matrix example: parallel cameras

\[ R = I \]

\[ T = [-d,0,0]^T \]

\[ E = [T_x]R = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix} \]

\[ p'^T E p = 0 \]

\[ \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0 \]
Essential matrix example: parallel cameras

\[ \mathbf{R} = \mathbf{I} \]
\[ \mathbf{T} = [\mathbf{T}_x] \mathbf{R} = \begin{pmatrix}
0 & 0 & 0 \\
0 & 0 & d \\
0 & -d & 0 \\
\end{pmatrix} \]

\[ \mathbf{p}'^T \mathbf{E} \mathbf{p} = 0 \]

\[
\begin{bmatrix}
x' & y' & f \\
0 & 0 & d \\
0 & -d & 0 \\
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
f \\
\end{bmatrix} = 0
\]

\[ \Leftrightarrow \begin{bmatrix}
x' & y' & f \\
0 & df & d \\
0 & -dy & -d \\
\end{bmatrix}
\begin{bmatrix}
0 \\
df \\
dy \\
\end{bmatrix} = 0 \]
Essential matrix example: parallel cameras

\[ \mathbf{R} = \mathbf{I} \]

\[ \mathbf{T} = [-d, 0, 0]^T \]

\[ \mathbf{E} = [\mathbf{T}_x] \mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix} \]

\[ \mathbf{p'}^T \mathbf{E} \mathbf{p} = 0 \]

\[ \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0 \]

\[ \Leftrightarrow \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 \\ df \\ -dy \end{bmatrix} = 0 \]

\[ \Leftrightarrow y = y' \]
Essential matrix example: parallel cameras

\[ R = I \]
\[ T = [-d, 0, 0]^T \]
\[ E = [T_x] R = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix} \]

\[ p'^T E p = 0 \]
\[
\begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0
\]

\[ \Leftrightarrow \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 \\ df \\ -dy \end{bmatrix} = 0 \]
\[ \Leftrightarrow y = y' \]

For the parallel cameras, image of any point must lie on same horizontal line in each image plane.
**Example I**: compute the fundamental matrix for a parallel camera stereo rig

\[ \mathbf{P} = K[I \mid 0] \quad \mathbf{P}' = K'[R \mid t] \]

\[ K = K' = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R} = \mathbf{I} \quad \mathbf{t} = \begin{pmatrix} tx \\ 0 \\ 0 \end{pmatrix} \]

\[ \mathbf{F} = K'^{-T} [\mathbf{t}] \times \mathbf{R} \mathbf{K}^{-1} \]

\[ \begin{bmatrix} \frac{1}{a} & 0 & 0 \\ 0 & \frac{1}{b} & 0 \\ -\frac{u_0}{a} & -\frac{v_0}{b} & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -tx \\ 0 & tx & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{a} & 0 & -\frac{u_0}{a} \\ 0 & \frac{1}{b} & -\frac{v_0}{b} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\frac{tx}{b} \\ 0 & \frac{tx}{b} & 0 \end{bmatrix} \]

\[ \mathbf{x}'^T \mathbf{F} \mathbf{x} = (x' \ y' \ 1) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0 \]

- reduces to \( y = y' \), i.e. raster correspondence (horizontal scan-lines)
**F is a rank 2 matrix**

The epipole \( e \) is the null-space vector (kernel) of \( F \) (**exercise**), i.e. \( Fe = 0 \)

In this case

\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & -1 \\
0 & 1 & 0 \\
\end{bmatrix}
\begin{bmatrix}
1 \\
0 \\
0 \\
\end{bmatrix} = 0
\]

so that

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

**Geometric interpretation?**
Image pair rectification

Goal: Simplify stereo matching by “warping” the images

Apply projective transformation so that epipolar lines correspond to horizontal scanlines
Image pair rectification

Goal: Simplify stereo matching by “warping” the images

Apply projective transformation so that epipolar lines correspond to horizontal scanlines

\[
\begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix} = H_e
\]

map epipole e to (1,0,0)
Image pair rectification

Goal: Simplify stereo matching by “warping” the images

Apply projective transformation so that epipolar lines correspond to horizontal scanlines

\[
\begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix} = He
\]

map epipole e to (1,0,0)
try to minimize image distortion
Image pair rectification

Goal: Simplify stereo matching by “warping” the images

Apply projective transformation so that epipolar lines correspond to horizontal scanlines

\[
\begin{bmatrix}
1 \\
0 \\
0
\end{bmatrix} = He
\]

map epipole \( e \) to \((1,0,0)\)

try to minimize image distortion

problem when epipole in (or close to) the image
Planar rectification

Image Transformations: Find homographies $H'$ and $H$ so that after transformations, $F^n$ becomes $F$ of parallel cameras:

$$H'^{-T}FH^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$
Planar rectification

Image Transformations: Find homographies $H'$ and $H$ so that after transformations, $F^n$ becomes $F$ of parallel cameras:

$$H'^{-T}FH^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$
Planar rectification

Image Transformations: Find homographies $H'$ and $H$ so that after transformations, $F^n$ becomes $F$ of parallel cameras:

$$H'^{-T}FH^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$
Planar rectification

Bring two views to standard stereo setup
(moves epipole to $\infty$)
(not possible when in/close to image)

Image Transformations: Find homographies $H'$ and $H$ so that after transformations, $F^n$ becomes $F$ of parallel cameras:

$$H'^{-\top}FH^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$
Algorithm Rectification

Following Trucco & Verri book pp. 159

- known $T$ and $R$ between cameras
- Rotate left camera so that epipole $e_l$ goes to infinity along horizontal axis
- Apply same rotation to right camera to recover geometry
- Rotate right camera by $R^{-1}$
- Adjust scale
The algorithm consists of four steps:

- Rotate the left camera so that the epipoles go to infinity along the horizontal axis.
- Apply the same rotation to the right camera to recover the original geometry.
- Rotate the right camera by $R$.
- Adjust the scale in both camera reference frames.

To carry out this method, we construct a triple of mutually orthogonal unit vectors $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$. Since the problem is underconstrained, we are going to make an arbitrary choice. The first vector, $\mathbf{e}_1$, is given by the epipole; since the image center is in the origin, $\mathbf{e}_1$ coincides with the direction of translation, or

$$
\mathbf{e}_1 = \frac{T}{\|T\|}.
$$

The only constraint we have on the second vector, $\mathbf{e}_2$, is that it must be orthogonal to $\mathbf{e}_1$. To this purpose, we compute and normalize the cross product of $\mathbf{e}_1$ with the direction vector of the optical axis, to obtain

$$
\mathbf{e}_2 = \frac{1}{\sqrt{T_x^2 + T_y^2}} \begin{bmatrix} -T_y, T_x, 0 \end{bmatrix}^T.
$$

The third unit vector is unambiguously determined as

$$
\mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2.
$$

It is easy to check that the orthogonal matrix defined as

$$
R_{rect} = \begin{pmatrix}
\mathbf{e}_1^T \\
\mathbf{e}_2^T \\
\mathbf{e}_3^T
\end{pmatrix}
$$

(7.22)

rotates the left camera about the projection center in such a way that the epipolar lines become parallel to the horizontal axis. This implements the first step of the algorithm. Since the remaining steps are straightforward, we proceed to give the customary algorithm:
Algorithm RECTIFICATION

The input is formed by the intrinsic and extrinsic parameters of a stereo system and a set of points in each camera to be rectified (which could be the whole images). In addition, Assumptions 1 and 2 above hold.

1. Build the matrix $R_{\text{rect}}$ as in (7.22);
2. Set $R_l = R_{\text{rect}}$ and $R_r = R R_{\text{rect}}$;
3. For each left-camera point, $p_l = [x, y, f]^T$ compute
   $$R_l p_l = [x', y', z']$$
   and the coordinates of the corresponding rectified point, $p'_l$, as
   $$p'_l = \frac{f}{z'} [x', y', z'].$$
4. Repeat the previous step for the right camera using $R_r$ and $p_r$.

The output is the pair of transformations to be applied to the two cameras in order to rectify the two input point sets, as well as the rectified sets of points.

From: Trucco & Verri, Introductory Techniques for 3-D Computer Vision, pp. 157-161
More elegant Solution

- Idea: Mapping epipole to infinity → \([1,0,0]^T\)
- Factorization of matrix \(F=SM\), where \(S\) is skew symmetric and \(M\) representing the required homography (projective transformation).
- Use SVD:

\[
F = UDV^T = UWZD'V^T = (UWU^T)(UZD'V^T) = SM
\]

where

\[
W = \begin{bmatrix}
0 & 1 & 0 \\
-1 & 0 & 0 \\
0 & 0 & 0
\end{bmatrix} ;
Z = \begin{bmatrix}
0 & -1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

http://www.robots.ox.ac.uk/~vgg/hzbook/hzbook2/clarification_rectification.pdf
Stereo matching with general camera configuration
Image pair rectification
Other Material /Code

- Epipolar Geometry, Rectification:

- Fusiello, Trucco & Verri: Tutorial, Matlab code etc:
  [http://profs.sci.univr.it/~fusiello/demo/rect/](http://profs.sci.univr.it/~fusiello/demo/rect/)
Run Example

Demo for stereo reconstruction (out of date):
http://mitpress.mit.edu/e-journals/Videre/001/articles/Zhang/CalibEnv/CalibEnv.html

Updated Webpages:

SFM Example:

Software:
Example: Zhengyou Zhang

Fundamental matrix between the two cameras:

\[
\begin{bmatrix}
5.2049303\times 10^{-7} & -2.9159992\times 10^{-5} & -2.5405448\times 10^{-4} \\
2.9090123\times 10^{-5} & 6.7508671\times 10^{-8} & -5.6408271\times 10^{-3} \\
-5.0032429\times 10^{-3} & 5.6687599\times 10^{-3} & 1.00\times 10^{-0}
\end{bmatrix}
\]
Points have been extracted using Harris corner detector, point matches via fundamental matrix $F$ and search along epipolar lines.
Point matches found by a correlation technique
3D reconstruction represented by a pseudo stereogram

Additional Materials: Forward Translating Cameras
**Example II:** compute $F$ for a forward translating camera

$$P = K[I | 0] \quad P' = K'[R | t]$$

$$K = K' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R = I \quad t = \begin{pmatrix} 0 \\ 0 \\ t_z \end{pmatrix}$$

$$F = K' - T[\mathbf{t}] \times RK^{-1}$$

$$= \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -t_z & 0 \\ t_z & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$
From $l' = Fx$ the epipolar line for the point $x = (x, y, 1)^T$ is

$$l' = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} -y \\ x \\ 0 \end{pmatrix}$$

The points $(x, y, 1)^T$ and $(0, 0, 1)^T$ lie on this line.
Summary: Properties of the Fundamental matrix

- $F$ is a rank 2 homogeneous matrix with 7 degrees of freedom.

- **Point correspondence:**
  
  if $x$ and $x'$ are corresponding image points, then $x'\mathbf{T}Fx = 0$.

- **Epipolar lines:**
  
  - $l' = Fx$ is the epipolar line corresponding to $x$.
  - $l = F^{-\mathbf{T}}x'$ is the epipolar line corresponding to $x'$.

- **Epipoles:**
  
  - $Fe = 0$.
  - $F^{-\mathbf{T}}e' = 0$.

- **Computation from camera matrices $P, P'$:**
  
  $P = K[I \mid 0]$, $P' = K'[R \mid t]$, $F = K'^{-\mathbf{T}}[t] \times RK^{-1}$
Goal: 3D from Stereo via Disparity Map

\( \text{image } I(x,y) \quad \text{Disparity map } D(x,y) \quad \text{image } I'(x',y') \)

\((x',y') = (x + D(x,y), y)\)

F&P
Chapter 11
Example: Stereo to Depth Map
Example: Stereo to Depth Map