“Two are better than one; because they have a good reward for their labor”

 Ecclesiastes 4:9
Epipolar Geometry

CSE 4392-5369 “Vision-based Robot Sensing, Localization and Control”

Dr. Gian Luca Mariottini, Ph.D.
Department of Computer Science and Engineering
University of Texas at Arlington

WEB: http://ranger.uta.edu/~gianluca/
A brief overview on camera localization algorithms

In the previous classes we focused on camera localization using image features and 3-D points.

Pinhole Camera Calibration/Localization

**Assumption:** Known 3-D points (>=6)
**Goal:** Estimate $K, (C_W R, C_W t)$ wrt \{W\}

**Goal:** Estimate $C_W R, C_W t$ wrt \{W\}

Stereo Camera Calibration/Localization

**Assumption:** No points >=3.
**Goal:** Estimate $(L_1 R, L_1 t)$. 

Pinhole Camera Calibration/Localization

**Assumption:** Known $K$ and 3-D planar pnts (>=4)
**Goal:** Estimate $(C_W R, C_W t)$ wrt \{W\}

**Assumption:** Two views, $K$ and $(u_i, u_i')$
**Goal:** Estimate $(C', R', C't)$

**Tool:** Multiple-View (Epipolar) geometry
Applications of Multiple-View Geometry

- Camera pose estimation and 3-D reconstruction from calibrated/uncalibrated views

PhotoTourism, Courtesy Snavely et al.

- Camera/Robot Localization
- Vision-Based Robot Navigation

Gian Luca Mariottini, Univ. Texas at Arlington
**Epipolar Geometry: definitions**

**Baseline:**
The baseline is the line joining two cameras' optical centers.

**Epipole:**
The epipole is the point of intersection (in pixels) of the baseline with the image plane. There are two epipoles, \( e \) and \( e' \), one for each image.

**Epipolar plane:**
An epipolar plane is the plane passing through the camera centers and each 3-D point.

**Epipolar line:**
An epipolar line is the line of intersection of the epipolar plane with the image plane. For each point in each image, there is one epipolar line.

- **Note that:**
  for different world points \( X \), the epipolar plane rotates about the baseline.
The epipolar line constrains the search for correspondence from a region to a line. If a point is observed in one image, then the location of the corresponding point in the other image must lie on the epipolar line.
We want to find an expression for the epipolar line.

\[ \begin{align*} 
    c'X & = c'R_cX + c't \\
    \text{and by taking the cross-product with } t \\
    c't \times c'X & = c't \times c'R_cX + c't \times c't \\
    \text{The cross product can be re-written using skew-symmetric matrices} \\
    [c't] \times c'X & = [c't] \times c'R_cX + [c't] \times c't \\
    [c't] \times c'X & = [c't] \times c'R_cX \\
    \end{align*} \]

In order to zero the part on the left of the equal sign...

\[ c'X^T([c't] \times c'X) = c'X^T([c't] \times c'R_cX) \]

thus finally leading to the **epipolar constraint**: 

\[ c'X^T[c't] \times c'R_cX = 0 \]
Note that the epipolar constraint \( c'X^T [c' t] \times c'RcX = 0 \) is also valid if we consider the rays \((c\mathbf{x}, c'\mathbf{x})\).

Remember that:
\[
\begin{align*}
  c\mathbf{X} &= \lambda c\mathbf{x} \\
  c'\mathbf{X} &= \lambda' c'\mathbf{x}
\end{align*}
\]

so that the epipolar constraint can be written as a function only of the rays in the image:

\[
  c'X^T [c' t] \times c'RcX = 0
\]

This constraint says that:

For a given point (ray) \( c\mathbf{x} \) in one image, the corresponding point (ray) \( c'\mathbf{x} \) in the other camera lies on a (epipolar) line that has equation (not referred in pixels):

\[
  \ell' \sim [c' t] \times c'RcX
\]
The matrix \([c'_t] \times c'_R\) is also called essential matrix:

\[ E \triangleq [c'_t] \times c'_R \]

Under the above definition the epipolar constraint becomes:

\[ c'_x^T E c_x = 0 \]

The essential matrix can be used to find the location of the epipoles:

\[
\lambda' \ c'_t = c'_R p_e + c'_t \quad \Rightarrow \quad [c'_t] \times c'_R p_e = 0 \quad \Rightarrow \quad E p_e = 0
\]

so, the epipole lies in the right nullspace of \( E \)

Analogously it can be proven that:

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

Note that \( p_e \) is a ray expressed in the camera frame \( \{C\} \) so it must be projected to the image plane to find the corresponding value of the epipole \( e \) in pixels.
Let us consider again the epipolar constraint: 

$$c' x^T E c x = 0$$

It will be very useful to rewrite as a constraint in the image space!

$$c x = K^{-1} \tilde{u}$$

$$c' x = K^{-1} \tilde{u}'$$

$$\Rightarrow \quad \tilde{u}'^T K^{-T} E K^{-1} \tilde{u} = 0$$

which finally becomes:

$$\tilde{u}'^T F \tilde{u} = 0$$

where \( F \triangleq K^{-T} E K^{-1} \) is also referred to as fundamental matrix.

The fundamental matrix allows us to finally define the epipolar lines (in the image plane)

$$\tilde{u}'^T \ell' = 0$$

where \( \ell' = F \tilde{u} \)

$$\Rightarrow \quad u'(1) + v' \ell'(2) + \ell'(3) = 0$$

$$v' = \left(-\ell'(1)/\ell'(2)\right) u' - \left(\ell'(3)/\ell'(2)\right)$$
The Epipolar Geometry: example of converging cameras

3 corner features in left image

Epipolar lines in right image

Epipolar lines in left image

3 corner features in right image
The Epipolar Geometry: example of parallel cameras

3 corner features in left image

Epipolar lines in right image

Epipolar lines in left image

3 corner features in right image
Applications of Epipolar Geometry in Robot Control

desired view

current view

baseline

O

O'

P=(X,Y,Z)

1st step

2nd step

Gian Luca Mariottini, Univ. Texas at Arlington
The eight-point algorithm: introduction

We have seen how corresponding image-points are related by the epipolar constraint.

\[ \tilde{u}' \,^T \, F \, \tilde{u} = 0 \quad \leftrightarrow \quad c' \, x^T \, E \, c \, x = 0 \]

\[ c \, x = K^{-1} \, \tilde{u} \]

\[ c' \, x = K^{-1} \, \tilde{u}' \]

Our GOAL is now this:

<< Use pairs of corresponding image points \((u_i, u'_i)\) in order to provide an estimate of the essential matrix \(E\) >>

In what follows, we will show that this is possible by observing (at least) 8 image points from a calibrated camera.

IMPORTANT: Points must not be from a planar object!
The epipolar constraint $c^T x^T E c x = 0$ can be equivalently re-written as:

$$[x_c x'_c, y_c x'_c, z_c x'_c, x_c y'_c, y_c y'_c, z_c y'_c, x_c z'_c, y_c z'_c, z_c z'_c] \equiv a_i^T \Rightarrow e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9 = 0$$

The above constraint $a_i^T x = 0$ is valid for each corresponding point-pair.

What is the minimum number of point-pairs we need to estimate $x$ uniquely?

8 is the minimum number of point pairs needed (theoretical number, no noise!)

$\begin{bmatrix} a_1^T & a_2^T & \ldots & a_8^T \end{bmatrix} x = 0 \Rightarrow [U, D, V] = \text{SVD}(A) \Rightarrow x = V(:, \text{end})$

- unstack the vector $x$ into the matrix $E$
- Eventually perform Step1) of the $E$ decomposition (see next slides)
Eight-point algorithm: Flow diagram

Convert to rays

Construct vectors $a_i^T$

Compute $x$ as $x = V(:, \text{end})$

Stack the vector $x$ into the matrix $E$
Let us assume that the intrinsic camera calibration $K$ is known and equal for both views.

Since $F \triangleq K^{-T}EK^{-1}$ we can then obtain:

$$E = K^TFK$$

Once $E$ is known, it is possible to decompose it in its two originating rotation matrix and translation vector.

**Step 1)** $[U, D, V] = \text{SVD}(E) \Rightarrow \bar{E} = U \text{diag}\{1, 1, 0\} V'$;

**Step 2)** $[U, D, V] = \text{SVD}(E) \Rightarrow [t]_\times = UR_{z, \pm \pi/2}DUT \cdot \text{sign}(\det(UVT))$

$$R = UR_{z, \pm \pi/2}^TV^T \cdot \text{sign}(\det(UVT))$$

**Step 3)** $[t]_\times \rightarrow t$

**Step 4)** Among the 4 possible solutions, keep the one with the 3-D reconstructed points in front of the camera.
Euclidean constraints and structure reconstruction

If we consider the translation “up-to-scale”...

\[ \lambda_i' c^i x_i = \lambda_i c^i R^c x_i + \gamma c^i t \]

Is it possible to show that, in an analogous way as before:

\[ M \lambda = 0 \]

where: \( \lambda = [\lambda_1, \lambda_2, \ldots, \lambda_n, \gamma]^T \)

and:

\[
M = \begin{bmatrix}
[ c^i x_1 ] \times [ c^i R^c x_1 ] & 0 & 0 & 0 & [ c^i x_1 ] \times [ c^i t ] \\
0 & \ddots & 0 & 0 & \vdots \\
0 & 0 & [ c^i x_{n-1} ] \times [ c^i R^c x_{n-1} ] & 0 & [ c^i x_{n-1} ] \times [ c^i t ] \\
0 & 0 & 0 & [ c^i x_n ] \times [ c^i R^c x_n ] & [ c^i x_n ] \times [ c^i t ]
\end{bmatrix}
\]

From this it is possible to obtain a 3-D reconstruction (up to scale)
**Planar scenes and homography**

- **The epipolar geometry** cannot be reliably estimated in the case of **planar scene**.

- The two views are related by a **rigid-body transformation** \((C'_C R, C'_C t)\) such that:

\[
c'_C X = c'_C R c_X + c'_C t
\]

- Any of the 3-D points lying on the plane \(\Pi\) satisfies at this **additional constraint**:

\[
n_1 X_c + n_2 Y_c + n_3 Z_c + d = 0
\]

where \(d\) is the distance of the plane from \(\{C\}\) and:

\[
c_n = [n_1, n_2, n_3]^T
\]

\[
c_X = [X_c, Y_c, Z_c]^T
\]

- So that the **co-planarity constraint** becomes:

\[
c_n^T c_X = -d
\]
Using equation (i) into the rigid-body transformation yields:

\[
c' X = c'R^c X + c't \cdot 1 = c'R^c X + c't \cdot \left(-\frac{c n^T}{d}^c X\right)
= \left(c'R - \frac{1}{d} c't^c n^T\right)^c X
\]

So we have obtained such a relationship:

\[
c' X = H^c X
\]

where the matrix \( H \) (3x3) is called planar homography matrix and is given by:

\[
H = c'R - \frac{1}{d} c't^c n^T
\]

Since \( c^c X = \lambda^c X \) and \( c' X = \lambda'^c c' X \) then the homography expression is equiv. to

\[
c' X \sim H c X
\]

Interesting application: map-based localization
Homography constraint: estimation

For any ray \( c \mathbf{x} \) in the first image, the corresponding ray \( c' \mathbf{x} \) is uniquely determined as:

\[
\begin{align*}
    c' \mathbf{x} & \sim H \, c \mathbf{x} \\
\end{align*}
\]

In order to eliminate the unknown scale we can multiply by \( [c' \mathbf{x}] \times \)

\[
[c' \mathbf{x}] \times H \, c \mathbf{x} = 0
\]

“Planar Epipolar Constraint”

After some calculations (try it!) we can rewrite this expression as:

\[
\begin{align*}
A_i \\
\end{align*}
\]

That can be solved solving for \( h \) in:

\[
Ah = 0
\]

where:

\[
A = \begin{bmatrix}
A_1 \\
\vdots \\
A_n
\end{bmatrix}
\]

with \( n \geq 4 \)
Homography estimation: examples