Question of the Day:

• How can we represent the relationships between cameras and the world?
The Book

Multiple View Geometry in computer vision

Richard Hartley and Andrew Zisserman

Why Camera Geometry

• To relate what we see in images to what is happening in the world
Vocabulary

“Canonical Position”: center at origin, Z axis is optical axis

The Camera Matrix

• Pinhole model represented as matrix

\[
\begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix} \mapsto \begin{pmatrix}
fX + Zp_x \\
fY + Zp_y \\
Z
\end{pmatrix} = \begin{bmatrix}
f & p_x & 0 \\
f & p_y & 0 \\
1 & 0 & 1
\end{bmatrix} \begin{pmatrix}
X \\
Y \\
Z \\
1
\end{pmatrix}
\]
Internal Camera Parameters

- Focal Length
- Principal point
- Pixel size
- Skew

\[
K = \begin{bmatrix}
\alpha_x & x_0 \\
\alpha_y & y_0 \\
1 & 1
\end{bmatrix}
\]

- Map from world to image coordinates, assuming a camera in canonical position

Camera Matrix in Canonical Position

- Map from world to image coordinates, assuming a camera in canonical position

\[
x = K[I | 0]X
\]
Camera Pose

• Translation

\[
P = K \begin{bmatrix} I & -\mathbf{C} \end{bmatrix}
\]

Position of camera center

Binocular Stereo

• Use image disparity to calculate depth

\[
u_1 = \frac{x_f}{Z} \quad \text{u2} = \frac{(x-t)f}{Z}
\]

\[
Z = \frac{tf}{(u_1-u_2)}
\]

Find matching \(u_1,u_2\)
Camera Pose

- Rotation

\[ P = KR[I \mid -\widetilde{C}] \]

(Ignore of) Camera Orientation

Position of camera center

Epipolar Geometry

- Relationship between cameras
Epipolar Lines

• Relationship between points in one image and lines in another

Epipolar Lines

• A point in one image defines a line in the other
• How to find (a matrix to give us) this line?
Epipolar Lines

- A point in one image defines a line in the other
- How to find (a matrix to give us) this line?

Back-projection of Image Points

- Point in an image defines a ray emanating out into the scene

How do you define a 3D line?
Back-projection of Image Points

• Point in an image defines a ray emanating out into the scene

\[ X(\lambda) = P^+x + \lambda C. \]

Points on this line are imaged in the second view as \( P' \ast X(\lambda) \)

How do you define a 3D line?

Vector from projecting image point back into scene

Pseudo-inverse for matrix that does not have full rank
Fundamental Matrix

- Matrix that expresses the relationship between camera images and maps from image points to epipolar lines
- All epipolar lines intersect at the epipole
- The epipolar line is a cross product between two points:
  - The epipole
  - Any point you can come up with that will be on the line based on image point

\[ \mathbf{F} = [\mathbf{e}'] \times \mathbf{P} \mathbf{P}'^+ \]

\[ \mathbf{L}' = \mathbf{F} \mathbf{x} \]

Epipolar Line Constraint

\[ \mathbf{x}'^T \mathbf{F} \mathbf{x} = 0. \]

- \( \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{y} + \mathbf{C} = 0 \)
- Given \( \mathbf{A}, \mathbf{B}, \mathbf{C} \) (\( \mathbf{F} \mathbf{x} \)), if you plug in a point on the line for \( \mathbf{x}, \mathbf{y} \), the answer will be zero

- Epipolar line constraint follows since image of \( \mathbf{X} \) must be on epipolar line
So what?

• How can I use this camera geometry to find out things about the world using images?

What can you do with camera geometry?

• Given 3D objects, find their image coordinates (booooorring)

• Given image points and 3D points, infer the pose of the camera

• Given image coordinates and camera matrices for multiple cameras, reconstruct the 3D point

• Perform multiple view data association for tracking

• Infer the movement of the camera (from image correspondences only)
Estimating a Linear Transform

• Rewrite the point transformation equation:

\[
\begin{bmatrix}
  x & y & 1 & 0 & 0 & 0 \\
  0 & 0 & 0 & x & y & 1 \\
\end{bmatrix}
\begin{bmatrix}
  a \\
  b \\
  c \\
  d \\
  e \\
  f \\
\end{bmatrix} =
\begin{bmatrix}
  x' \\
  y' \\
\end{bmatrix}
\]

How to estimate [a b c d e f]?

Estimating a Linear Transformation

• DLT (Direct Linear Transformation)

• We will use it several times to infer things about camera geometry:
  – Reconstruct 3D point
  – Recover pose of camera
  – Estimate fundamental matrix
DLT Basic Idea

• Set up a linear system that should be equal to zero (i.e. by taking a cross product of two vectors) : $Ah = 0$
• Minimize the norm: $||Ah||$
• By taking the eigenvector of $A^TA$ corresponding to the smallest eigenvalue

Using DLT to compute a homography

• Given, image correspondences, $x$ and $x'$
• Want: homography / transformation $H$
• Set up cross product of $x \times H \times x'$
• Factor out elements of $H$ into a vector with a system of equations like $Ah = 0$
• $h$ is the eigenvector of $A^TA$ with smallest eigenvalue
Using DLT to reconstruct a 3D point

• Given: image points $x$ and $x'$, camera matrices $P$ and $P'$
• Want 3D point such that $x = PX$ and $x' = P'X$
• Set up cross product of $x \times PX$ and $x' \times P'X$
• Factor out $X$
• Get system of equations of the form $AX = 0$
• $X$ is eigenvector of $A^TA$ with least eigenvalue

Estimate 3D point

• Given image points and camera matrices, estimate the 3D world point

$$AX = 0$$
can then be composed, with

$$A = \begin{bmatrix} xP^3 - P^1T \\ yP^3 - P^2T \\ x'P'^3 - P'^1T \\ y'P'^3 - P'^2T \end{bmatrix}$$
Estimating the Camera Matrix

• Need a calibration object

\[ P = KR[I \mid -\tilde{C}] \]

• Given: 3D points X and corresponding image points [x y w]

• Obtain: Camera matrix, P

Note: Not factored in a nice way!

Factoring the Camera Matrix

• Matrices can be factored in many ways

• From P, we can read out matrix “M”

• We know that we want M factored into a rotation (orthonormal) matrix and an upper triangular matrix (K)

• The way to do this is called “RQ decomposition”

\[ P = KR[I \mid -\tilde{C}] \]

\[ K[R \mid -R\tilde{C}] \]

\[ [M \mid -M\tilde{C}] \]


Fundamental Matrix

- Relationship between cameras
- Maps from points in one image to lines in the other image
- Epipolar line constraint:

\[ x'^T F x = 0. \]

Estimating the Fundamental Matrix

- Use epipolar line constraint
- Factor out elements of F
- F is eigenvector of \( A^T A \) with least eigenvalue

\[
Af = \begin{bmatrix}
    x'_1 x_1 & x'_1 y_1 & x'_1 \ x'_1 & y'_1 x_1 & y'_1 y_1 & y'_1 & x_1 & y_1 & 1 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x'_n x_n & x'_n y_n & x'_n \ x'_n & y'_n x_n & y'_n y_n & y'_n & x_n & y_n & 1 \\
\end{bmatrix} f = 0
\]
Estimating the Fundamental Matrix

• Use epipolar line constraint
  – Need to scale image coordinates to get good results!
  – Center of image points at origin, magnitude = \sqrt{2}

\[
Af = \begin{bmatrix}
    x'_1 x'_1 & x'_1 y'_1 & x'_1 & y'_1 x'_1 & y'_1 y'_1 & y'_1 & x'_1 & y'_1 & 1 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    x'_n x'_n & x'_n y'_n & x'_n & y'_n x'_n & y'_n y'_n & y'_n & x'_n & y'_n & 1 \\
\end{bmatrix} f = 0
\]

Essential Matrix

• (One thing) we really would like: Recover camera pose from image correspondences only
• Need to factor out camera calibration matrices: K, K'

\[
\hat{x} = K^{-1} x. \text{ Then } \hat{x} = [R \mid t] x.
\]
\[
\hat{x}'^T E \hat{x} = 0
\]
\[
E = K'^T F K.
\]
Recover Camera Pose from Essential Matrix

- Most important to know: It can be done

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

SVD of \( E \) is \( U \text{diag}(1, 1, 0)V^T \).

\[ P' = [UWV^T | +u_3] \text{ or } [UWV^T | -u_3] \]
\[ \text{or } [UW^TV^T | +u_3] \text{ or } [UW^TV^T | -u_3]. \]

Recover Camera Pose from Essential Matrix

- Most important to know:
  Four possible options (Need to test)

\[ P' = [UWV^T | +u_3] \]
\[ \text{or } [UWV^T | -u_3] \]
\[ \text{or } [UW^TV^T | +u_3] \]
\[ \text{or } [UW^TV^T | -u_3]. \]
Radial Distortion

• Before you can estimate anything, you need your images to be square

• [http://toothwalker.org/optics.html](http://toothwalker.org/optics.html)

Radial Distortion

• Cause:
  Restricting light to the image sensor at location other than the lens

[http://toothwalker.org/optics.html](http://toothwalker.org/optics.html)
Radial Distortion

- Complex effect, need to pick a simple model
- Most common/popular is polynomial

\[
\begin{align*}
\hat{x} &= x_c + L(r)(x - x_c) \\
\hat{y} &= y_c + L(r)(y - y_c), \\
L(r) &= 1 + \kappa_1 r + \kappa_2 r^2 + \kappa_3 r^3 + \ldots
\end{align*}
\]

- Given coefficients, you can correct the distortion

Radial Distortion

- Estimating Coefficients is a bit involved
  [http://www.vision.caltech.edu/bouguetj/calib-doc/](http://www.vision.caltech.edu/bouguetj/calib-doc/)
Here it is in song form:

• The Fundamental Matrix Song:

    http://danielwedge.com/fmatrix/