Multiple View Geometry in Computer Vision

Prasanna Sahoo
Department of Mathematics
University of Louisville
• In our last lecture, we examined the action of the projection matrix on planes, lines, conics and quadrics and developed their forward and back-projection properties.

• We also examined the importance of camera center. The effect of zooming by a factor $k$ is to multiply the matrix $K$ on the right by $\text{diag}(k, k, 1)$. 
• In this lecture, we will continue on the topic “the importance of the camera center” and then look at the camera calibration problem using the image of the absolute conic (IAC).

• Recall that an absolute conic is an imaginary conic on the plane at infinity and thus it is not visible.
Camera Rotation

Suppose the camera is rotated about its center with no changes in the internal parameters.
If $x$ and $x'$ are the images of a point $X$ before and after a pure rotation, then we have

$$x = K[I|0]X \quad \text{and} \quad x' = K[R|0]X.$$ 

Hence

$$x' = K[R|0]X = KR[I|0]X = KRK^{-1}x = Hx$$

where $H := KRK^{-1}$ known as conjugate rotation.
Between images (a) and (b) the camera rotates about the camera center. Similarly, between images (a) and (c) the camera rotates about the camera center and translates.

image (a)  image (b)  image (c)
Example 2. Consider the images (a) and (b) of the last slide. The homography $H$ has eigenvalues \( \{\mu, \mu e^{i\theta}, \mu e^{-i\theta}\} \), where $\mu$ is a scale factor. The eigenvalues of $H$ are same as that of $R$.

- The angle of rotation $\theta$ between views can be computed from the phase of the complex eigenvalues.

- Between the images (a) and (b) the angle of rotation is estimated as $4.66^\circ$. 
• The vanishing point of the rotation axis can be computed from the eigenvector of $H$ corresponding to the real eigenvalue.

• The axis vanishing point is estimated as

$$(-0.0088, 1, 0.0001)^T.$$  

• Axis vanishing point is at infinity in the $y$ direction.
Some Applications

The homographic relation between images with same camera center can be exploited in several ways:

- To create synthetic image by projective warping.
- To create a panoramic mosaic.
New images corresponding to different camera orientation can be generated from an existing image by warping with planar homographies.
Planar panoramic mosaicing

- Choose one image of the set as a reference.
- Compute $\mathbf{H}$ which maps one of the other images of the set to this reference image.
- Projectively warp the image with this homography.
- Repeat the last two steps.
Planar panoramic mosaicing
Reduced Camera Matrix

If we choose the coordinates of world points as

\[ X_1 = (1,0,0,0)^T \quad X_2 = (0,1,0,0)^T \]
\[ X_3 = (0,0,1,0)^T \quad X_4 = (0,0,0,1)^T \]

and the coordinates of image points as

\[ x_1 = (1,0,0)^T \quad x_2 = (0,1,0)^T \]
\[ x_3 = (0,0,1)^T \quad x_4 = (1,1,1)^T \]
then the camera matrix

\[
P = \begin{pmatrix}
a & 0 & 0 & -d \\
0 & b & 0 & -d \\
0 & 0 & c & -d \\
\end{pmatrix}
\]

satisfies \( x_i = P X_i \) for \( i = 1, 2, 3, 4 \).

Further, it is easy to see that

\[
P (a^{-1}, b^{-1}, c^{-1}, d^{-1})^T \begin{pmatrix}
a^{-1} \\
b^{-1} \\
c^{-1} \\
d^{-1} \\
\end{pmatrix} = 0.
\]
Note that if $\mathbf{P} \mathbf{C} = 0$, then $\mathbf{C}$ is the camera center.

- This shows that $(a^{-1}, b^{-1}, c^{-1}, d^{-1})^T$ is the camera center $\mathbf{C}$.

- This matrix $\mathbf{P}$ is called the reduced camera matrix and it is specified by the 3 degrees of freedom of the camera center.
Moving the Camera Center

• If the camera center remains fixed then all cameras gather the same image content.

• So same camera center implies same image, just a rearrangement (that is just a planar projection in $\mathbb{P}^3$).

• Thus camera center $C$ must move to change the image content. No zooming, warping, rotation can change this.
• How can we determine, from image alone, whether the camera center has moved or not?

Consider two 3-space points $X_1$ and $X_2$ which have coincident images in the first view.
If the camera center is moved to $C'$, then the image coincidence is lost. This relative displacement of previously coincident image points is called **parallax**.

The vector between $x'_1$ and $x'_2$ is the **parallax**.

The line through $x'_1$ and $x'_2$ is the **epipolar line**.
Between image (a) and image (b) there is no motion parallax.

image (a)  
image (b)
Between image (a) and image (c) there is a motion parallax.
Camera Calibration

- A image point $x$ back-projects to a ray defined by $x$ and the camera center $C$.
- Camera calibration relates the image point to ray's direction.
Suppose $D = (x_d, y_d, z_d, 0)^T$ is a point at infinity and let $d = (x_d, y_d, z_d)^T$.

Let the camera matrix $P$ be

$$
P = \begin{pmatrix}
f & 0 & px & 0 \\
0 & f & py & 0 \\
0 & 0 & 1 & 0
\end{pmatrix} = \begin{pmatrix}
f & 0 & px \\
0 & f & py \\
0 & 0 & 1
\end{pmatrix} \begin{bmatrix} I | 0 \end{bmatrix} = K \begin{bmatrix} I | 0 \end{bmatrix}.
$$

Then $x = PD = K \begin{bmatrix} I | 0 \end{bmatrix} D = K d$.

Hence the ray through the image point $x$ has the direction $d = K^{-1} x$. 
**Result 1.** The camera calibration matrix $K$ is the affine transformation between $x$ and the ray's direction $d = K^{-1}x$ measured in the camera’s Euclidean’s coordinate frame.
Angle between $C$ and two image points

The angle $\theta$ between two rays.
The angle between the camera center $C$ and two rays with direction $d_1$ and $d_2$ corresponding to image points $x_1$ and $x_2$ is given by

$$\cos(\theta) = \frac{d_1^T d_2}{\sqrt{(d_1^T d_1)(d_2^T d_2)}}$$

$$= \frac{x_1^T (K^{-T} K^{-1}) x_2}{\sqrt{(x_1^T (K^{-T} K^{-1}) x_1)(x_2^T (K^{-T} K^{-1}) x_2)}}$$
• If the matrix $K$ is known, then the matrix $K^{-T}K^{-1}$ is also known.

• Thus the angle between rays can be measured from their corresponding image points.

• A camera is said to be **calibrated** if the calibration matrix $K$ is known.

• A calibrated camera is direction sensor (or serves as protractor).
The image line $L$ and the camera center $C$ defines a scene plane.

The calibration matrix $K$ can be used to find a relation between an image line and a scene plane.
Result 1. An image line $L$ defines a plane $\Pi$ through the camera center with normal direction $n = K^T L$.

Proof:

- Let $x$ be a point on the line $L$.
- Let $n$ be the normal to the plane $\Pi$.
- The line $L$ will back-project to directions $d = K^{-1} x$.
- These directions are orthogonal to $n$.
- Hence $d^T n = 0$. This implies $x^T K^{-T} n = 0$. 
• Each point \(x\) on \(L\) satisfies \(x^\top L = 0\).

• \(x^\top L = 0\) and \(x^\top K^{-\top}n = 0\) imply \(L = K^{-\top}n\).

• Hence \(n = K^\top L\).

This result is important and will be used many times.
What is $K^{-T}K^{-1}$?

Recall that

- Plane at infinity $\Pi_\infty$ holds all “horizon points” $d$.
- Absolute conic $\Omega_\infty$ is imaginary outermost circle of the plane at $\Pi_\infty$.

Next we will show that $K^{-T}K^{-1}$ is the image of the absolute conic (IAC).
Result 2. Suppose \( P = KR[I \mid -\bar{C}] \) is a general camera matrix. Then \( K^{-T}K^{-1} \) is a conic, and

\[
K^{-T}K^{-1} = P\Omega_{\infty}.
\]

Proof:

• Let \( X_{\infty} \) be a point on \( \Pi_{\infty} \). Then \( X_{\infty} = (d^T, 0)^T \).
• Hence \( x = P X_{\infty} = KR[I \mid -\bar{C}] (d^T, 0)^T = KRd \).
• So \( KR \) is a planar homography between \( \Pi_{\infty} \) and \( x \).
• Under the point homography \( x \mapsto Hx \) the conic \( C \) maps as \( C \xrightarrow{\omega} H^{-T}CH^{-1} \).
• $\Omega_\infty$ is a conic in $\Pi_\infty$, therefore $C = \Omega_\infty = I$.

• Hence

$$\omega = H^{-T}CH^{-1} = (KR)^{-T}I(KR)^{-1}$$

$$= \left((KR)^{T}\right)^{-1} (KR)^{-1}$$

$$= \left(R^{T}K^{T}\right)^{-1} R^{-1} K^{-1}$$

$$= K^{-T} R^{-T} R^{-1} K^{-1}$$

$$= K^{-T} K^{-1} \quad \text{since } R \text{ is orthogonal.}$$

• So $P\Omega_\infty = \omega = K^{-T} K^{-1}$. 
Few Remarks.

- IAC $\omega$ depends only on $K$.
- Angle between two rays is unchanged under a projective transformation of the image. (see p.200)
- The dual absolute conic $\omega^* = \omega^{-1} = KK^T$.
- The dual absolute conic $\omega^*$ is the image of $Q_{\infty}^*$.
- Once the matrix $\omega$ is known $K$ can be determined by Cholesky factorization.
• A plane $\Pi$ intersects $\Pi_\infty$ in a line $\ell_\infty$ and this line intersects $\Omega_\infty$ in two points which are circular points of $\Pi$. Image of circular points lie on $\omega$ at the points at which the vanishing line of the plane $\Pi$ intersects the IAC $\omega$.

• IAC is a ‘magic tool’ for camera calibration.

• With IAC, find $P$ matrix from an image of just 3 (non-coplanar) squares.
Cholesky factorization

If $\omega$ is a symmetric positive definite matrix, then $\omega$ can be uniquely decomposed into a product of an upper-triangular matrix $K$ and its transpose. That is,

$$\omega = KK^T.$$ 

MATLAB has a built-in function `chol` which finds the Cholesky factorization.
• Take a non-singular matrix $A$ given by

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 2 & 1 \\ 3 & 1 & 3 \end{pmatrix}.$$  

• Find a symmetric positive definite matrix $B$ by taking $B = A^T A$.

• Invoke built-in function $\text{chol}$ by typing $K = \text{chol}(B)$ on MATLAB prompt.

• MATLAB gives $K$ as $A =$

$$A = \begin{pmatrix} 3.7417 & 2.4054 & 3.7417 \\ 0 & 1.7928 & 1.1155 \\ 0 & 0 & 1.9379 \end{pmatrix}.$$
Example 1. A simple Calibration Device.

The image of three squares can be used to compute the calibration matrix $K$.

Recall that on plane $\Pi$ there are two circular points given by $(1, i, 0)^T$ and $(1, -i, 0)^T$. 
• For each square compute the homography $\mathbf{H}$ that maps its corner points $(0, 0)^T$, $(1, 0)^T$, $(0, 1)^T$, $(1, 1)^T$ to their imaged points.

• Apply this $\mathbf{H}$ to circular points for the plane $\Pi$ of that square to obtain $\mathbf{H}(1, i, 0)^T$ and $\mathbf{H}(1, -i, 0)^T$.

• Points $\mathbf{H}(1, i, 0)^T$ and $\mathbf{H}(1, -i, 0)^T$ are on IAC $\omega$. 
• Fit a conic $\omega$ to the six (only needs five) imaged circular points using the equation

$$\begin{pmatrix}
  x_1^2 & x_1 y_1 & y_1^2 & x_1 & y_1 & 1 \\
  x_2^2 & x_2 y_2 & y_2^2 & x_2 & y_2 & 1 \\
  x_3^2 & x_3 y_3 & y_3^2 & x_3 & y_3 & 1 \\
  x_4^2 & x_4 y_4 & y_4^2 & x_4 & y_4 & 1 \\
  x_5^2 & x_5 y_5 & y_5^2 & x_5 & y_5 & 1 \\
\end{pmatrix} \omega = 0.$$

• Compute the calibration matrix $K$ from $\omega = K^{-T} K^{-1}$ using the Cholesky factorization.
(a) Calibration device

(b) Calibration matrix $K$

$$
\begin{pmatrix}
1108.3 & -9.8 & 525.8 \\
0 & 1097.8 & 395.9 \\
0 & 0 & 1
\end{pmatrix}
$$
END