Optical center

World origin

\((R, t)\)
Normalized image plane

World origin

Optical center

\( (0,0) \)

\( Z_c = 1 \)
Sensor plane
(0,0)

Normalized image plane

Optical center

World origin

$X_w$

$Y_w$

$Z_w$
Sensor plane

Normalized image plane

Optical center

World origin
Sensor plane
(0,0)

Normalized image plane

Optical center

World origin

X_w

Y_w

Z_w

P

P_{sensor}

P_{normalized}

X_c

Y_c

Z_c

K
Pose estimation

- Single image of points with known 3D locations
- Estimate the rotation and translation (extrinsic parameters) of the camera

\[ p_i = (X_i, Y_i, Z_i, W_i) \]

\[ \theta_{ij} \]

\[ x_i \]

\[ x_j \]

\[ d_i \]

\[ d_j \]

\[ d_{ij} \]

Figure 6.4 Pose estimation by the direct linear transform and by measuring visual angles and distances between pairs of points.

Given a set of corresponding (D and )D points \( \{ (\hat{x}_i, p_i) \} \) where the \( \hat{x}_i \) are unit directions obtained by transforming (D pixel measurements \( x_i \) to unit norm )D directions \( \hat{x}_i = N(K^{-1}x_i) = K^{-1}x_i / ||K^{-1}x_i|| \)

the unknowns are the distances \( d_i \) from the camera origin \( c \) to the )D points \( p_i \)

The cosine law for triangle \( \Delta(c, p_i, p_j) \) gives us

\[ f_{ij}(d_i, d_j) = d_i^2 + d_j^2 - 2d_i d_j c_{ij} = 0 \]

where \( c_{ij} = \cos \theta_{ij} = \hat{x}_i \cdot \hat{x}_j \)

\[ d_{ij}^2 = ||p_i - p_j||^2 \]

We can take any triplet of constraints \( (f_{ij}, f_{ik}, f_{jk}) \) and eliminate the \( d_j \) and \( d_k \) using Sylvester resultants to obtain a quartic equation in \( d_i^2 \)

\[ g_{ijk}(d_i^2) = a_4 d_i^8 + a_3 d_i^6 + a_2 d_i^4 + a_1 d_i^2 + a_0 = 0 \]

Given five or more correspondences we can generate \((n-1)(n-2)/2\) triplets to obtain a linear estimate using SVD for the values of \( (d_8^i, d_6^i, d_4^i, d_2^i) \)

Quan and Lan, Estimates for
Triangulation

- Known relationships between two or more cameras
- Find the 3D point that minimizes distance to all camera rays
Epipolar geometry

- Describes the relationship between projection of a 3D point in two camera images
- Can be used to recover rotation and translation between cameras
- Or to search for correspondences between two images

![Diagram of epipolar geometry]

\[
\begin{align*}
\hat{x}_0^T R \times (c_1 - c_0) &= \hat{x}_1^T R \times (c_1 - c_0) \\
\text{since the right hand side is a triple product with two identical entries} &
\end{align*}
\]

Another way to say this is that the cross product matrix \([t] \times \) is skew symmetric and returns 0 when pre- and post-multiplied by the same vector.

We therefore arrive at the basic epipolar constraint

\[
\hat{x}_0^T E \hat{x}_0 = 0,
\]

where \(E = [t] \times R\) is called the essential matrix.

An alternative way to derive the epipolar constraint is to notice that in order for the cameras to be oriented so that the rays \(\hat{x}_0\) and \(\hat{x}_1\) intersect in \(p\), the vectors connecting the two camera centers \(c_1 - c_0 = -R^{-1}t\) and the rays corresponding to pixels \(x_0\) and \(x_1\) at \(p\) must be co-planar. This requires that the triple product

\[
(\hat{x}_0, R^{-1} \hat{x}_1, -t) = (\hat{x}_1, \hat{x}_1, -t) = \hat{x}_1 \cdot (\hat{x}_0^T (t \times R \hat{x}_0)) = \hat{x}_1^T (t \times R \hat{x}_0) = 0.
\]

Notice that the essential matrix \(E\) maps a point \(\hat{x}_0\) in image 0 into a line \(l_1 = E \hat{x}_0\) in image 1, since \(\hat{x}_0^T l_1 = 0\). Figure 7.3 (30 All such lines must pass through the second epipole \(e_1\) which is therefore defined as the left singular vector of \(E\) with a 0 singular value or equivalently the projection of the vector \(t\) into image 0. The dual transpose of these...