<table>
<thead>
<tr>
<th></th>
<th>Structure (scene geometry)</th>
<th>Motion (camera geometry)</th>
<th>Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pose Estimation</td>
<td>known</td>
<td>estimate</td>
<td>3D to 2D correspondences</td>
</tr>
<tr>
<td>Triangulation</td>
<td>estimate</td>
<td>known</td>
<td>2D to 2D correspondences</td>
</tr>
<tr>
<td>Reconstruction</td>
<td>estimate</td>
<td>estimate</td>
<td>2D to 2D correspondences</td>
</tr>
</tbody>
</table>
Given a single image, estimate the exact position of the photographer
Pose estimation for digital display

Touch-Consistent Perspective for Direct Interaction under Motion Parallax

Yusuke Sugano, Kazuma Harada and Yoichi Sato
Institute of Industrial Science, The University of Tokyo
3D Pose Estimation
(Resectioning, Calibration, Perspective n-Point)

Given a set of matched points

\[ \{ X_i, x_i \} \]

point in 3D space \hspace{1cm} point in the image

and camera model

\[ x = f(X; p) = PX \]

projection model \hspace{1cm} parameters \hspace{1cm} Camera matrix

Find the (pose) estimate of

\[ P \]
Recall: Camera Models (projections)

We'll use a perspective camera model for pose estimation
What is Pose Estimation?

Given

\[ O_{\text{camera}} \rightarrow O_{\text{image}} \rightarrow X_{\text{world point}} \]

\[ O_{\text{world}} \]
What is Pose Estimation?

What is Pose Estimation?

- \( f \) = ?
- \( R \) = ?
- \( t \) = ?
- \( \boldsymbol{X} \) = world point

Estimate

- intrinsic parameters
- extrinsic parameters

What is Pose Estimation?
Same setup as homography estimation using DLT
(slightly different derivation here)
Mapping between 3D point and image points

\[
\begin{bmatrix}
    x \\
    y \\
    z
\end{bmatrix}
= \begin{bmatrix}
    p_1 & p_2 & p_3 & p_4 \\
    p_5 & p_6 & p_7 & p_8 \\
    p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\begin{bmatrix}
    X \\
    Y \\
    Z \\
    1
\end{bmatrix}
\]

What are the unknowns?

\[
\begin{bmatrix}
    x \\
    y \\
    z
\end{bmatrix}
= \begin{bmatrix}
    \quad p_1^\top & \quad \\
    \quad p_2^\top & \quad \\
    \quad p_3^\top & \quad
\end{bmatrix}
\begin{bmatrix}
    X
\end{bmatrix}
\]

Inhomogeneous coordinates

\[
x' = \frac{p_1^\top X}{p_3^\top X} \quad y' = \frac{p_2^\top X}{p_3^\top X}
\]

(non-linear correlation between coordinates)

*How can we make these relations linear?*
How can we make these relations linear?

\[ x' = \frac{p_1^T X}{p_3^T X} \quad y' = \frac{p_2^T X}{p_3^T X} \]

Make them linear with algebraic manipulation…

\[ p_2^T X - p_3^T X y' = 0 \]

\[ p_1^T X - p_3^T X x' = 0 \]

Now you can setup a system of linear equations with multiple point correspondences (this is just DLT for different dimensions)
\[ \begin{align*}
  p_2^\top X - p_3^\top X y' &= 0 \\
  p_1^\top X - p_3^\top X x' &= 0
\end{align*} \]

In matrix form …

\[
\begin{bmatrix}
  X^\top & 0 & -x' X^\top \\
  0 & X^\top & -y' X^\top
\end{bmatrix}
\begin{bmatrix}
  p_1 \\
p_2 \\
p_3
\end{bmatrix} = 0
\]

For N points …

\[
\begin{bmatrix}
  X_1^\top & 0 & -x' X_1^\top \\
  0 & X_1^\top & -y' X_1^\top \\
  \vdots & \vdots & \vdots \\
  X_N^\top & 0 & -x' X_N^\top \\
  0 & X_N^\top & -y' X_N^\top
\end{bmatrix}
\begin{bmatrix}
  p_1 \\
p_2 \\
p_3
\end{bmatrix} = 0
\]
Solve for camera matrix by

\[ \hat{x} = \arg \min_x \|Ax\|^2 \text{ subject to } \|x\|^2 = 1 \]

\[ A = \begin{bmatrix}
X_1^\top & 0 & -x'X_1^\top \\
0 & X_1^\top & -y'X_1^\top \\
\vdots & \vdots & \vdots \\
X_N^\top & 0 & -x'X_N^\top \\
0 & X_N^\top & -y'X_N^\top
\end{bmatrix} \quad x = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \]

SVD!
Solve for camera matrix by

\[ \hat{x} = \arg \min_x \| A x \|^2 \text{ subject to } \| x \|^2 = 1 \]

\[
A = \begin{bmatrix}
X_1^\top & 0 & -x' X_1^\top \\
0 & X_1^\top & -y' X_1^\top \\
\vdots & \vdots & \vdots \\
X_N^\top & 0 & -x' X_N^\top \\
0 & X_N^\top & -y' X_N^\top \\
\end{bmatrix}
\]

\[
x = \begin{bmatrix}
p_1 \\
p_2 \\
p_3 \\
\end{bmatrix}
\]

Solution \( x \) is the column of \( V \) corresponding to smallest singular value of

\[ A = U \Sigma V^\top \]
Solve for camera matrix by

\[ \hat{x} = \arg \min_x \|Ax\|^2 \text{ subject to } \|x\|^2 = 1 \]

\[ A = \begin{bmatrix}
X_1^\top & 0 & -x'X_1^\top \\
0 & X_1^\top & -y'X_1^\top \\
\vdots & \vdots & \vdots \\
X_N^\top & 0 & -x'X_N^\top \\
0 & X_N^\top & -y'X_N^\top \\
\end{bmatrix} \]

\[ x = \begin{bmatrix}
p_1 \\
p_2 \\
p_3 \\
\end{bmatrix} \]

Equivalently, solution \( x \) is the Eigenvector corresponding to smallest Eigenvalue of \( A^\top A \).
Almost there …

\[
\mathbf{P} = \begin{bmatrix}
p_1 & p_2 & p_3 & p_4 \\
p_5 & p_6 & p_7 & p_8 \\
p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\]

*How do you get the intrinsic and extrinsic parameters from the projection matrix?*
Decomposition of the Camera Matrix

\[ P = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \]
Decomposition of the Camera Matrix

\[
P = \begin{bmatrix}
p_1 & p_2 & p_3 & p_4 \\
p_5 & p_6 & p_7 & p_8 \\
p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\]

\[
P = K[R|t]
\]
Decomposition of the Camera Matrix

\[
P = \begin{bmatrix}
p_1 & p_2 & p_3 & p_4 \\
p_5 & p_6 & p_7 & p_8 \\
p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\]

\[
P = K[R|t]
\]

\[
= K[R - Rc]
\]

\[
= [M - Mc]
\]
Decomposition of the Camera Matrix

\[ P = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \]

\[ P = K[R|t] \]
\[ = K[R| - Rc] \]
\[ = [M| - Mc] \]

Find the camera center \( C \)

Find intrinsic \( K \) and rotation \( R \)
Decomposition of the Camera Matrix

\[ P = \begin{bmatrix}
  p_1 & p_2 & p_3 & p_4 \\
  p_5 & p_6 & p_7 & p_8 \\
  p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix} \]

\[ P = K[R|t] \]
\[ = K[R| - Rc] \]
\[ = [M| - Mc] \]

Find the camera center \( \mathbf{c} \)
\[ P_c = 0 \]

Find intrinsic \( K \) and rotation \( R \)
Decomposition of the Camera Matrix

\[
P = \begin{bmatrix}
p_1 & p_2 & p_3 & p_4 \\
p_5 & p_6 & p_7 & p_8 \\
p_9 & p_{10} & p_{11} & p_{12}
\end{bmatrix}
\]

\[
P = K[R|t]
\]

\[
= K[R - Rc]
\]

\[
= [M - Mc]
\]

Find the camera center \( \mathbf{C} \)

\[
Pc = 0
\]

SVD of \( P \!

\( c \) is the Eigenvector corresponding to smallest Eigenvalue

Find intrinsic \( K \) and rotation \( R \)
Decomposition of the Camera Matrix

$$\mathbf{P} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix}$$

$$\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$$

$$= \mathbf{K}[\mathbf{R} - \mathbf{Rc}]$$

$$= [\mathbf{M} - \mathbf{Mc}]$$

Find the camera center $\mathbf{C}$

$$\mathbf{Pc} = 0$$

SVD of $\mathbf{P}$!

$c$ is the Eigenvector corresponding to smallest Eigenvalue

Find intrinsic $\mathbf{K}$ and rotation $\mathbf{R}$

$$\mathbf{M} = \mathbf{KR}$$

right upper triangle

orthogonal
Decomposition of the Camera Matrix

\[ \mathbf{P} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \]

\[ \mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}] \]
\[ = \mathbf{K}[\mathbf{R} - \mathbf{Rc}] \]
\[ = [\mathbf{M} - \mathbf{Mc}] \]

Find the camera center \( \mathbf{C} \)
\[ \mathbf{Pc} = 0 \]
SVD of \( \mathbf{P}! \)
\( \mathbf{c} \) is the Eigenvector corresponding to smallest Eigenvalue

Find intrinsic \( \mathbf{K} \) and rotation \( \mathbf{R} \)
\[ \mathbf{M} = \mathbf{KR} \]
RQ decomposition
Simple AR program

1. Compute point correspondences (2D and AR tag)
2. Estimate the pose of the camera $P$
3. Project 3D content to image plane using $P$

3D locations of planar marker features are known in advance

3D content prepared in advance
Do you need computer vision to do this?
<table>
<thead>
<tr>
<th></th>
<th>Structure (scene geometry)</th>
<th>Motion (camera geometry)</th>
<th>Measurements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pose Estimation</strong></td>
<td>known</td>
<td>estimate</td>
<td>3D to 2D correspondences</td>
</tr>
<tr>
<td><strong>Triangulation</strong></td>
<td>estimate</td>
<td>known</td>
<td>2D to 2D correspondences</td>
</tr>
<tr>
<td><strong>Reconstruction</strong></td>
<td>estimate</td>
<td>estimate</td>
<td>2D to 2D correspondences</td>
</tr>
</tbody>
</table>