Camera Calibration, Binocular Stereo, Part 2
Multiview Stereo
Epipolar Geometry
Methods for Binocular Scene Reconstruction

Lecture by Margrit Betke, CS 585, March 21, 26, & 28, 2024
Camera Transformation Problems

1. Interior Orientation = Camera Calibration = Intrinsic Calibration:
   What kind of camera?
   Simple version: Find focal length $f$ and principal point $p$ (= point where optical axis intersects image plane)
   Better: Correct for lens distortion, check if angle between $x$ & $y$ axes is $90^\circ$

2. Exterior Orientation = Extrinsic Calibration = Hand-Eye Calibration in Robotics:
   Where is the camera? Find center of projection of camera, and orientation of camera coordinate system in world coordinate system

3. Absolute Orientation = Alignment of 2 Cameras or 2 Medical Scans
   Find relationship between cameras. 3D coordinates of points are known

4. Relative Orientation = Alignment of 2 Cameras
   Find relationship between cameras. 3D coordinates not known, only rays known
Camera Transformation Problems: Unknown rotation $R$ & translation $r_0$

Transformation equation: $R r_{\text{camera}} + t = r_{\text{world}}$

2. Exterior Orientation = Extrinsic Calibration = Hand-Eye Calibration in Robotics:
   Where is the camera? Find center of projection of camera and orientation of camera coordinate system in world coordinate system

   Transformation equation: $R r_{\text{left}} + r_0 = r_{\text{right}}$

3. Absolute Orientation = Alignment of 2 Cameras or 2 Medical Scans
   Find relationship between cameras. 3D coordinates of points are known

4. Relative Orientation = Alignment of 2 Cameras
   Find relationship between cameras. 3D coordinates not known, only rays known
How can we represent rotation?

• Euler angles: roll, yaw, pitch (3 degrees of freedom)
• Quaternions (more later, to prepare, review imaginary numbers)
• Axis and angle: Axis is a unit vector $\omega$ (2 degrees of freedom)

  Angle: $\theta$

  Rodriguez’ Formula:

  $$r' = r \cos \theta + (\hat{\omega} \cdot r) \hat{\omega} (1 - \cos \theta) + (\hat{\omega} \times r) \sin \theta$$

$$\omega \times r = \begin{pmatrix}
\omega_2 r_3 - \omega_3 r_2 \\
\omega_3 r_1 - \omega_1 r_3 \\
\omega_1 r_2 - \omega_2 r_1
\end{pmatrix}$$
Most popular representation: Rotation Matrix

Derivation of Rotation Matrix:

Rotation in the image plane by angle $\theta$:

Original Position:

\[
\begin{pmatrix}
  x_1 \\
  y_1 \\
  z_1
\end{pmatrix}
=\begin{pmatrix}
  r \cos \alpha_1 \\
  r \sin \alpha_1 \\
  1
\end{pmatrix}
\]

Rotated Position:

\[
\begin{pmatrix}
  x_2 \\
  y_2 \\
  z_2
\end{pmatrix}
=\begin{pmatrix}
  r \cos \alpha_2 \\
  r \sin \alpha_2 \\
  1
\end{pmatrix}
\]

\[\alpha_2 = \alpha_1 + \theta\]
\[ \alpha_2 = \alpha_1 + \theta \]

\[
\begin{pmatrix}
  x_2 \\
  y_2 \\
  z_2
\end{pmatrix}
= \begin{pmatrix}
  r \cos \alpha_2 \\
  r \sin \alpha_2 \\
  1
\end{pmatrix}
\]

\[
= \begin{pmatrix}
  r \cos(\alpha_1 + \theta) \\
  r \sin(\alpha_1 + \theta) \\
  1
\end{pmatrix}
= \begin{pmatrix}
  r \cos \alpha_1 \cos \theta - r \sin \alpha_1 \sin \theta \\
  r \sin \alpha_1 \cos \theta + r \cos \alpha_1 \sin \theta \\
  1
\end{pmatrix}
\]

\[
= \begin{pmatrix}
  x_1 \cos \theta - y_1 \sin \theta \\
  y_1 \cos \theta + x_1 \sin \theta \\
  1
\end{pmatrix}
\]

\[
= \begin{pmatrix}
  \cos \theta & -\sin \theta & 0 \\
  \sin \theta & \cos \theta & 0 \\
  0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
  x_1 \\
  y_1 \\
  1
\end{pmatrix}
\]

\[ R^T R = RR^T = I \]

R is an orthonormal matrix = columns (or rows) add up to 1 and are perpendicular to each other (dot product = 0)
Camera Transformation Problems: Unknown rotation $R$ & translation $r_0$

Transformation equation:  \[ R \begin{pmatrix} r_{\text{camera}} \\ t \end{pmatrix} + r_0 = r_{\text{world}} \]

2. Exterior Orientation = Extrinsic Calibration = Hand-Eye Calibration in Robotics:
   Where is the camera? Find center of projection of camera and orientation of camera coordinate system in world coordinate system

   Transformation equation:  \[ R \begin{pmatrix} r_{\text{left}} \\ r_0 \end{pmatrix} = r_{\text{right}} \]

3. Absolute Orientation = Alignment of 2 Cameras or 2 Medical Scans
   Find relationship between cameras. 3D coordinates of points are known

4. Relative Orientation = Alignment of 2 Cameras
   Find relationship between cameras. 3D coordinates not known, only rays known
Relative Orientation for Binocular Stereo

Goal: Recovery of position and orientation of one imaging system relative to another from correspondences between rays

Given: 2D coordinates of image points of same world point
Special Case

Baseline $b = \text{Distance between Center of Projections (CoPs)}$

$Z = \frac{bf}{\delta}$
Relative Orientation = General Binocular Stereo

Use perspective projection equations:

\[
\begin{align*}
\frac{x_{\text{left}}}{f_{\text{left}}} &= \frac{X_{\text{left}}}{Z_{\text{left}}} \\
\frac{y_{\text{left}}}{f_{\text{left}}} &= \frac{Y_{\text{left}}}{Z_{\text{left}}} \\
\frac{x_{\text{right}}}{f_{\text{right}}} &= \frac{X_{\text{right}}}{Z_{\text{right}}} \\
\frac{y_{\text{right}}}{f_{\text{right}}} &= \frac{Y_{\text{right}}}{Z_{\text{right}}}
\end{align*}
\]

Transformation equation: \( R \ r_{\text{left}} + r_0 = r_{\text{right}} \)  
\( R \) = rotation matrix, \( r_0 \) = translation
Relative Orientation for Binocular Stereo

Transformation equation: $R \mathbf{r}_{\text{left}} + \mathbf{r}_0 = \mathbf{r}_{\text{right}}$

Unknown: Rotation matrix $R$, translation $\mathbf{r}_0$, Z coordinates of $\mathbf{r}_{\text{left}}$, $\mathbf{r}_{\text{right}}$
Relative Orientation for Binocular Stereo

\[ R \mathbf{r}_{\text{left}} + r_0 = \mathbf{r}_{\text{right}} \]

is equivalent to:

\[ r_{11} X_{\text{left}} + r_{12} Y_{\text{left}} + r_{13} Z_{\text{left}} + r_{14} = X_{\text{right}} \]
\[ r_{21} X_{\text{left}} + r_{22} Y_{\text{left}} + r_{23} Z_{\text{left}} + r_{24} = Y_{\text{right}} \]
\[ r_{31} X_{\text{left}} + r_{32} Y_{\text{left}} + r_{33} Z_{\text{left}} + r_{34} = Z_{\text{right}} \]

Insert Perspective Projection Equations:

\[ x_{\text{left}}/f = X_{\text{left}}/Z_{\text{left}} \quad y_{\text{left}}/f = Y_{\text{left}}/Z_{\text{left}} \]
\[ x_{\text{right}}/f = X_{\text{right}}/Z_{\text{right}} \quad y_{\text{right}}/f = Y_{\text{right}}/Z_{\text{right}} \]

\[ r_{11} x_{\text{left}} Z_{\text{left}}/f + r_{12} y_{\text{left}} Z_{\text{left}}/f + r_{13} Z_{\text{left}} + r_{14} = x_{\text{right}} Z_{\text{right}}/f \]
\[ r_{21} x_{\text{left}} Z_{\text{left}}/f + r_{22} y_{\text{left}} Z_{\text{left}}/f + r_{23} Z_{\text{left}} + r_{24} = y_{\text{right}} Z_{\text{right}}/f \]
\[ r_{31} x_{\text{left}} Z_{\text{left}}/f + r_{32} y_{\text{left}} Z_{\text{left}}/f + r_{33} Z_{\text{left}} + r_{34} = Z_{\text{right}} \]

Multiply by \( f/Z_{\text{left}} \)
Relative Orientation for Binocular Stereo

\[ R \mathbf{r}_{\text{left}} + \mathbf{r}_0 = \mathbf{r}_{\text{right}} \]

is equivalent to:

\[ r_{11} x_{\text{left}} + r_{12} y_{\text{left}} + r_{13} f + r_{14} f/Z_{\text{left}} = x_{\text{right}} Z_{\text{right}}/Z_{\text{left}} \]
\[ r_{21} x_{\text{left}} + r_{22} y_{\text{left}} + r_{23} f + r_{24} f/Z_{\text{left}} = y_{\text{right}} Z_{\text{right}}/Z_{\text{left}} \]
\[ r_{31} x_{\text{left}} + r_{32} y_{\text{left}} + r_{33} f + r_{34} f/Z_{\text{left}} = f Z_{\text{right}}/Z_{\text{left}} \]

One measurement pair \((x_{\text{left}}, y_{\text{left}})\) and \((x_{\text{right}}, y_{\text{right}})\) => 3 equations

with 14 unknowns \(r_{11}, r_{12}, ..., r_{34}\), and \(Z_{\text{right}}, Z_{\text{left}}\)
Relative Orientation

One measurement pair \((x_{\text{left}}, y_{\text{left}})\) and \((x_{\text{right}}, y_{\text{right}})\) \(\Rightarrow\) 3 equations

with 12 unknowns \(r_{11}, r_{12}, \ldots, r_{34}\) and 2 unknowns \(Z_{\text{right}}, Z_{\text{left}}\)

Trick: To solve for 14 unknowns:

Use \(n\) measurements \(\Rightarrow\) \(3n\) equations

Find additional equations
Relative Orientation

One measurement pair \((x_{\text{left}}, y_{\text{left}})\) and \((x_{\text{right}}, y_{\text{right}})\) => 3 equations with 12 unknowns \(r_{11}, r_{12}, \ldots, r_{34}\), and 2 unknowns \(Z_{\text{right}}, Z_{\text{left}}\)

One extra equation:

Scale factor ambiguity \(r_0, Z_{\text{right}}, Z_{\text{left}}\)

\[\iff kr_0, kZ_{\text{right}}, kZ_{\text{left}}\]

Force \(r_0\) to be unit vector

\[\Rightarrow |r_0|=1\]
Relative Orientation

One measurement pair \((x_{\text{left}}, y_{\text{left}})\) and \((x_{\text{right}}, y_{\text{right}})\) => 3 equations
with 14 unknowns \(r_{11}, r_{12}, \ldots, r_{34}\) and \(Z_{\text{right}}, Z_{\text{left}}\)

\[
\begin{align*}
  r_{11} x_{\text{left}} + r_{12} y_{\text{left}} + r_{13} f + r_{14} f/ Z_{\text{left}} &= x_{\text{right}} Z_{\text{right}}/ Z_{\text{left}} \\
  r_{21} x_{\text{left}} + r_{22} y_{\text{left}} + r_{23} f + r_{24} f/ Z_{\text{left}} &= y_{\text{right}} Z_{\text{right}}/ Z_{\text{left}} \\
  r_{31} x_{\text{left}} + r_{32} y_{\text{left}} + r_{33} f + r_{34} f/ Z_{\text{left}} &= f Z_{\text{right}}/ Z_{\text{left}}
\end{align*}
\]

# unknowns: 12 for \(R, r_0\)

2n for \(Z_{\text{right}}, Z_{\text{left}}\) for each of \(n\) pairs of measurements

12 + 2n unknowns
Relative Orientation

One measurement pair \((x_{\text{left}}, y_{\text{left}})\) and \((x_{\text{right}}, y_{\text{right}})\) \(\Rightarrow 3\) equations

with 14 unknowns \(r_{11}, r_{12}, ..., r_{34}\), and \(Z_{\text{right}}, Z_{\text{left}}\)

Number of equations: 6 for orthonormal \(R\) (columns sum to 1, dot products 0)

1 for unit length translation: \(|r_0|=1\)

3n for 3 equations per measurement pair

7 + 3n equations
Relative Orientation

One measurement pair \((x_{\text{left}}, y_{\text{left}})\) and \((x_{\text{right}}, y_{\text{right}})\) => 3 equations
with 14 unknowns \(r_{11}, r_{12}, \ldots, r_{34}\), and \(Z_{\text{right}}, Z_{\text{left}}\)

\[
\begin{align*}
    r_{11} x_{\text{left}} + r_{12} y_{\text{left}} + r_{13} f + r_{14} f/Z_{\text{left}} &= x_{\text{right}} Z_{\text{right}}/Z_{\text{left}} \\
    r_{21} x_{\text{left}} + r_{22} y_{\text{left}} + r_{23} f + r_{24} f/Z_{\text{left}} &= y_{\text{right}} Z_{\text{right}}/Z_{\text{left}} \\
    r_{31} x_{\text{left}} + r_{32} y_{\text{left}} + r_{33} f + r_{34} f/Z_{\text{left}} &= f Z_{\text{right}}/Z_{\text{left}}
\end{align*}
\]

# unknowns: 12 for \(R, r_0\)
2n for \(Z_{\text{right}}, Z_{\text{left}}\) for each of \(n\) pairs of measurements

Number of equations: 6 for orthonormal \(R\) (columns sum to 1, dot products 0)
1 for unit length translation \(r_0\)
3n for 3 equations per measurement pair

Need at least \(n = ?\) measurement pairs: 12 + 2 * \(n\) = 7 + 3*n
Relative Orientation

One measurement pair \((x_{\text{left}}, y_{\text{left}})\) and \((x_{\text{right}}, y_{\text{right}})\) => 3 equations

with 14 unknowns \(r_{11}, r_{12}, \ldots, r_{34}\), and \(Z_{\text{right}}, Z_{\text{left}}\)

\[
\begin{align*}
    r_{11} x_{\text{left}} + r_{12} y_{\text{left}} + r_{13} f + r_{14} f/Z_{\text{left}} &= x_{\text{right}} Z_{\text{right}}/Z_{\text{left}} \\
    r_{21} x_{\text{left}} + r_{22} y_{\text{left}} + r_{23} f + r_{24} f/Z_{\text{left}} &= y_{\text{right}} Z_{\text{right}}/Z_{\text{left}} \\
    r_{31} x_{\text{left}} + r_{32} y_{\text{left}} + r_{33} f + r_{34} f/Z_{\text{left}} &= f Z_{\text{right}}/Z_{\text{left}} \\
\end{align*}
\]

\# unknowns: 12 for \(R, r_0\)

2n for \(Z_{\text{right}}, Z_{\text{left}}\) for each of \(n\) pairs of measurements

Number of equations: 6 for orthonormal \(R\) (columns sum to 1, dot products 0)

1 for unit length translation \(|r_0|=1\)

3n for 3 equations per measurement pair

Need at least 5 measurement pairs: 12 + 2 * 5 = 22 = 7 + 3*5
Relative Orientation

\[
\begin{align*}
r_{11} x_{\text{left}} + r_{12} y_{\text{left}} + r_{13} f + r_{14} f/Z_{\text{left}} &= x_{\text{right}} Z_{\text{right}}/Z_{\text{left}} \\
r_{21} x_{\text{left}} + r_{22} y_{\text{left}} + r_{23} f + r_{24} f/Z_{\text{left}} &= y_{\text{right}} Z_{\text{right}}/Z_{\text{left}} \\
r_{31} x_{\text{left}} + r_{32} y_{\text{left}} + r_{33} f + r_{34} f/Z_{\text{left}} &= f Z_{\text{right}}/Z_{\text{left}}
\end{align*}
\]

\( n \) measurement pairs \((x_{\text{left}}, y_{\text{left}})\) and \((x_{\text{right}}, y_{\text{right}})\)  \( \Rightarrow \)  \( 3n \) equations + 7

with 14 unknowns \( r_{11}, r_{12}, \ldots, r_{34}, \) and \( Z_{\text{right}}, Z_{\text{left}} \)

Need at least 5 measurement pairs --

Does that mean 5 pairs are enough?
Relative Orientation

14 unknowns \( r_{11}, r_{12}, \ldots, r_{34}, \) and \( Z_{\text{right}}, Z_{\text{left}} \)

Need at least 5 measurement pairs

Does that mean 5 pairs are enough? No – the equations are not linear

No – there is likely noise involved

Nonetheless: Computer Vision courses and textbooks make this look like a linear problem that can be solved using a few measurement pairs. Methods such as the 8-point algorithm, or computing the “fundamental matrix,” are sensitive to noise and numerically unstable. They are not used in practice. But the math is elegant...
The idea to use homogeneous coordinates (first used in projective geometry in 1827) for computer vision comes from computer graphics. Note that task of computer graphics generally is to create images, and of computer vision to interpret images, i.e., inverse tasks.

In computer graphics, using homogeneous coordinates is convenient because operations such as rotation, scaling, translation, and perspective projection can be represented as matrices. A sequence of such operations can be represented as a sequence of matrix multiplications, enabling fast processing. Using Cartesian coordinates, perspective projection and translation cannot be expressed as matrix multiplications.
Recall: Perspective Projection

Projection Equation:
\[ x = f \frac{X}{Z} \]
\[ y = f \frac{Y}{Z} \]
"Elegant" Computer Vision Math: Projective Geometry

Cartesian coordinates (=“heterogeneous” coordinates):
Image point \((x,y)^T = (fX/Z, fY/Z)^T\)

Homogeneous coordinates add a dimension 2D->3D, 3D->4D:
Image point \((x,y,w)^T = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{pmatrix} (X,Y,Z,1)^T = (X, Y, Z/f)^T\)

Image point \((x,y,w)^T = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} (X,Y,Z,1)^T = (fX, fY, Z)^T\)

Both map back to \((x,y)^T\)
Projective Geometry: Projection Matrix and PP Shift

Projection matrix:
\[
\begin{pmatrix}
  f & 0 & 0 & 0 \\
  0 & f & 0 & 0 \\
  0 & 0 & 1 & 0
\end{pmatrix}
\]

Principal Point shifted \((p_x, p_y)^T\):
\[
\begin{pmatrix}
  f & 0 & p_x & 0 \\
  0 & f & p_y & 0 \\
  0 & 0 & 1 & 0
\end{pmatrix} = \begin{pmatrix}
  f & 0 & p_x \\
  0 & f & p_y \\
  0 & 0 & 1
\end{pmatrix} \begin{pmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 1 & 0
\end{pmatrix} = K \begin{pmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 1 & 0
\end{pmatrix}
\]

\(K = \begin{pmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 1 & 0
\end{pmatrix}\)

General 3D to 2D perspective projection with same image & camera coordinate origins, \(z=1\)

2D to 2D transformation accounting for shift \(p\) and focal length \(f\)
Projective Geometry: Projection Matrix and PP Shift

Projection matrix:
\[
\begin{pmatrix}
f & 0 & 0 & 0 \\
0 & f & 0 & 0 \\
0 & 0 & 1 & 0 \\
\end{pmatrix}
\]

Principal Point shifted \((p_x, p_y)^T\):
\[
\begin{pmatrix}
f & 0 & p_x & 0 \\
0 & f & p_y & 0 \\
0 & 0 & 1 & 0 \\
\end{pmatrix} = \begin{pmatrix}
f & 0 & p_x \\
0 & f & p_y \\
0 & 0 & 1 \\
\end{pmatrix} \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
\end{pmatrix} = K \begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\]

Camera Calibration
= Intrinsic Calibration
= Find \(K\)
= Find pp and \(f\)

\(K\) = 2D to 2D transformation accounting for shift \(p\) and focal length \(f\)

General 3D to 2D perspective projection with same image & camera coordinate origins, \(z=1\)
Camera Transformation Problems

1. **Interior Orientation = Camera Calibration = Intrinsic Calibration:**
   - What kind of camera?
   - Simple version: Find focal length $f$ and principal point $p$ (= point where optical axis intersects image plane)
   - Better: Correct for lens distortion, check if angle between x & y axes is $90^\circ$

2. **Exterior Orientation = Extrinsic Calibration = Hand-Eye Calibration in Robotics:**
   - Where is the camera? Find center of projection of camera, and orientation of camera coordinate system in world coordinate system

3. **Absolute Orientation = Alignment of 2 Cameras or 2 Medical Scans**
   - Find relationship between cameras. 3D coordinates of points are known

4. **Relative Orientation = Alignment of 2 Cameras**
   - Find relationship between cameras. 3D coordinates not known, only rays known
Camera Transformation Problems

1. Interior Orientation = Camera Calibration = Intrinsic Calibration:
   - What kind of camera?
   - Simple version: Find focal length $f$ and principal point $p$ (= point where optical axis intersects image plane)
   - Better: Correct for lens distortion, check if angle between $x$ & $y$ axes is $90^0$

2. Exterior Orientation = Extrinsic Calibration = Hand-Eye Calibration in Robotics:
   - Where is the camera? Find center of projection of camera, and orientation of camera coordinate system in world coordinate system

   Transformation equation: \( R \, r_{\text{camera}} + t = r_{\text{world}} \)
Camera Transformation Problems

Exterior Orientation = Extrinsic Calibration = Hand-Eye Calibration in Robotics:
Where is the camera? Find center of projection of camera, and orientation of camera coordinate system in world coordinate system

Transformation equation: \[ R \ r_{camera} + t = r_{world} \]
\[ R \ r_{camera} = r_{world} - t \]
\[ R^T R \ r_{camera} = R^T (r_{world} - t) \]
\[ r_{camera} = R^T (r_{world} - t) \]

R here is \( R^T \)
Camera Transformation Problems

Exterior Orientation = Extrinsic Calibration = Hand-Eye Calibration in Robotics:
Where is the camera? Find center of projection of camera, and orientation of camera coordinate system in world coordinate system

Transformation equation: \[ \mathbf{R} \mathbf{r}_{\text{camera}} + \mathbf{t} = \mathbf{r}_{\text{world}} \]

In homogeneous coordinates:
\[ \begin{pmatrix} \mathbf{R}^T \tilde{\mathbf{t}} \\ 0 \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{r}}_{\text{camera}} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} \mathbf{C}^W_2C \tilde{\mathbf{r}}_{\text{world}} \end{pmatrix} \]

\( \mathbf{R} \) here is \( \mathbf{R}^T \)

heterogeneous (= ‘regular’) 3D vectors \( \mathbf{r}_{\text{camera}}, \mathbf{r}_{\text{world}}, \mathbf{t} \)

homogeneous 4D vectors \( \tilde{\mathbf{r}}_{\text{camera}}, \tilde{\mathbf{r}}_{\text{world}}, \tilde{\mathbf{t}} \)
Projective Geometry: Mapping World Coordinates to Image Coordinates

\[
\tilde{r}_{\text{image}} = \begin{pmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tilde{r}_{\text{camera}} = \begin{pmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} R & -Rt \\ 0 & 1 \end{pmatrix} \tilde{r}_{\text{world}}
\]

\[
\tilde{r}_{\text{image}} = K \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} C^{W2C} \tilde{r}_{\text{world}}
\]

\[
\tilde{r}_{\text{image}} = P \tilde{r}_{\text{world}}
\]

(also written as \(\pi(\tilde{r}_{\text{world}})\) for projection)

Warning: This notation is dangerous... This is NOT a linear equation.

See also: Horn’s “Projective Geometry Considered Harmful”
https://people.csail.mit.edu/bkph/articles/Harmful.pdf
Triangulation: Computing World Coordinates

Assumptions:

- At least 2 cameras
- Intrinsic camera parameters are known (f, pp)
- Extrinsic camera parameters are known (mapping from each camera to the world coordinate system or mapping from one camera to the other)

Using these parameters, we can plug into the binocular stereo eq’s to compute 3D coordinates of $r_{\text{world}}$ from a matching pair of image points $r_{\text{image1}}$ & $r_{\text{image2}}$

If no error: $|\pi_i (r_{\text{world}}) - r_{\text{image,i}}| = 0$

But likely errors, so use a least squares approach:

$$r_{\text{world,best}} = \arg\min \sum |\pi_i (r_{\text{world}}) - r_{\text{image,i}}|^2$$
Stereoscopic 3D Reconstruction with Triangulation

3D position
of bat

left camera
right camera

Lisa Premerlani, 2007
Camera Transformation Problems:

Transformation equation: \[ R \mathbf{r}_{camera} + \mathbf{t} = \mathbf{r}_{world} \]

2. Exterior Orientation = Extrinsic Calibration = Hand-Eye Calibration in Robotics:
   Where is the camera? Find center of projection of camera and orientation of camera coordinate system in world coordinate system

   Transformation equation: \[ R \mathbf{r}_{left} + \mathbf{r}_0 = \mathbf{r}_{right} \]

4. Relative Orientation = Alignment of 2 Cameras
   Find relationship between cameras. 3D coordinates not known, only rays known

If 2. or 4. are solved, you can use triangulation to compute 3D points
Methods to Solve the Problem of General Binocular Stereo Reconstruction = Relative Orientation

• Longuet-Higgins’ 8-point Algorithm (1981):

\[(x_{left}, y_{left}, 1)^T F (x_{right}, y_{right}, 1) = 0\]

F is called the 3x3 “fundamental matrix” (use homogeneous coordinates)
Algorithm is sensitive to how accurate point pairs were located ( = numerically unstable)

• Variations of the 8-point Algorithm
  e.g. Hartley’s Normalized 8-point algorithm (1997)

• **Horn's Iterative Relative Orientation Method, 1990**. Does not use homogeneous coordinates

• Bundle Adjustment: Bundles of light rays, originating from 3D points, used to adjust estimates of camera parameters and depths
Multi-Camera Stereo

View from above:
Z axis = direction of gravity

i₁ is projection of scene point P₁ in view of camera C₁
i₂ is projection of scene point P₂ in view of camera C₁
Multi-Camera Stereo

\[ \begin{align*}
\text{i}_1 & \text{ could be projection of scene point } \text{G}_1 \\
\text{i}_2 & \text{ could projection of scene point } \text{G}_2
\end{align*} \]
3rd Camera resolves the ambiguity:
G₁ and G₂ are “ghosts” (non-existing points)
P₁ and P₂ are the true scene points
Green line is ray from $P_1$ into camera $C_3$. It appears as an “epipolar line” in the image of camera $C_1$. 
Discuss with your neighbor:

- What does the orange line in C1 represent?
- What do the green/red lines in C2 represent?
- What do the red/orange lines in C3 represent?
- Why do the lines in C1, C2, and C3 intersect in $i_1, i_3, \text{ and } i_5$?
The green line is the ray from $P_1$ into the 3rd camera. The orange line is the ray from $P_1$ into the 2nd camera. They appear as “epipolar lines” in the image of camera $C_1$ and must intersect at the same image point $i_1$. 
The green and red epipolar lines in the camera $C_2$ intersect at image point $i_3$.
The orange and red epipolar lines in the camera $C_3$ intersect at image point $i_5$. 

$P_1$ is imaged in the intersection
How to use epipolar lines for bat tracking:
Temporal Calibration

Used a lighter to register the two cameras in time
Epipolar Geometry

left image

right image

Image Credit: OpenCV.org
Epipolar Geometry

Formal Definition:
1. All possible scene points $M$ (M', M'', ...) that produce image $m_{\text{left}}$ are on a half line through $m_{\text{left}}$ and CoP$_{\text{left}}$.
2. All possible images $m_r$ of $M$ are images of this half line called "epipolar line."
3. The image of CoP$_{\text{left}}$ in the right image plane is called "epipole" i.e., e$_{\text{right_epipole}}$.

$3D$ scene point $M$

Baseline $b = \text{Distance between Center of Projections (CoPs)}$
Using Epipolar Geometry to Estimate Camera Motion

Sideway Camera Motion:

Epipole = Vanishing Point

Forward Camera Motion:

Epipole = Focus of Expansion

Fig. 9.7. Under a pure translational camera motion, 3D points appear to slide along parallel rails. The images of these parallel lines intersect in a vanishing point corresponding to the translation direction. The epipole e is the vanishing point.

Image Credit: Hartley & Zisserman, 2004
Remember from Linear Algebra:

• The dot product of two perpendicular vectors is zero.
• The cross product of two co-planar vectors computes a vector perpendicular to the plane the vectors span.
• The vector cross product can be expressed as the product of a skew-symmetric matrix and a vector: \( \mathbf{t} \times \mathbf{b} = [\mathbf{t}]_{x} \mathbf{b} \)

\[
[\mathbf{t}]_{x} = \begin{pmatrix}
0 & -t_3 & t_2 \\
t_3 & 0 & -t_1 \\
-t_2 & t_1 & 0 \\
\end{pmatrix}
\]
Derivation of the “Fundamental Matrix:"

These three vectors are in the same plane:
- \( t \), \( r_{\text{camera, left}} \), \( r_{\text{camera, right}} \)

Here both 3D heterogeneous or 4D homogeneous coordinates can be used.

\[
P_{\text{left}} = K[I|0] \]

\[
P_{\text{right}} = K[R|t] \]

\[
r_{\text{camera, left}}
\]

\[
r_{\text{camera, right}} = R (r_{\text{camera, left}} - t) \text{ or } (r_{\text{camera, left}} - t)^T = r_{\text{camera, right}}^T R
\]
Derivation of the “Fundamental Matrix:"

\[ P_{\text{left}} = K[I|0] \]
\[ P_{\text{right}} = K[R|t] \]

Epipolar lines:
\[ \tilde{L}_{\text{right}} = E \tilde{r}_{\text{camera, left}} \]
\[ \tilde{L}_{\text{left}} = E^{T} \tilde{r}_{\text{camera, right}} \]

Epipoles:
\[ E \tilde{e}_{\text{left}} = 0 \]
\[ \tilde{e}_{\text{right}}^{T} E = 0 \]

Here use 4D homogeneous coordinates

\[ \tilde{r}_{\text{camera, right}}^{T} (R^{T} [t]_{x}) \tilde{r}_{\text{camera, left}} = 0 \]

“Essential matrix” \( E \)

\[ \tilde{r}_{\text{camera, right}}^{T} E \tilde{r}_{\text{camera, left}} = 0 \]

“Fundamental matrix” \( F \)

\[ \tilde{r}_{\text{image, right}}^{T} K_{\text{right}}^{-T} E K_{\text{left}}^{-1} \tilde{r}_{\text{image, left}} = 0 \]

\[ \tilde{r}_{\text{image, right}}^{T} F \tilde{r}_{\text{image, left}} = 0 \]

\[ \tilde{e}_{\text{right}}^{T} E = 0 \]

\[ \tilde{t}^{T} E = 0 \]

\( F \) is of rank 2
Longuet-Higgins’ 8-point Algorithm (1981):

\[(x_{\text{left}}, y_{\text{left}}, 1)^T F (x_{\text{right}}, y_{\text{right}}, 1) = 0\]

F is called the 3x3 “fundamental matrix” (use homogeneous coordinates)

Algorithm is sensitive to how accurate point pairs were located (= numerically unstable)

• Variations of the 8-point Algorithm
  
  e.g. Hartley’s Normalized 8-point algorithm (1997)

• **Horn's Iterative Relative Orientation Method, 1990.** Does not use homogeneous coordinates

• Bundle Adjustment: Bundles of light rays, originating from 3D points, used to adjust estimates of camera parameters and depths
Methods to Solve the Problem of General Binocular Stereo Reconstruction

Longuet-Higgins’ 8-point Algorithm (1981):

\[
\begin{align*}
\begin{pmatrix}
\tilde{r}_{\text{camera, right}}^T \\
\tilde{r}_{\text{camera, left}}
\end{pmatrix} F &= \begin{pmatrix}
f_{11} & f_{12} & f_{13} \\
\vdots & \ddots & \vdots \\
f_{31} & f_{32} & f_{33}
\end{pmatrix} \begin{pmatrix}
x_l \\
y_l \\
1
\end{pmatrix} = 0
\end{align*}
\]

\[
\begin{pmatrix}
x_r x_l, x_r y_l, x_r y_l x_l, y_r x_l, y_r y_l, y_r, x_l, y_l, 1
\end{pmatrix} = 0
\]

Algorithm is sensitive to how accurate point pairs were located ( = numerically unstable)
Longuet-Higgins’ 8-point Algorithm for Binocular Stereo Reconstruction

Use 8 matching points in both views to create matrix $U$:

$$Uf = \begin{pmatrix} x_r x_l & x_r y_l & x_r & y_r x_l & y_r y_l & y_r & x_l & y_l & 1 \\ \vdots \\ \vdots \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix} = 0$$

Compute $f$ as $\arg \min_{\|f\|=1} \|Uf\|^2$ by finding an Eigenvector of $U^TU$.

Result likely produces a matrix $F$ that is not singular. Trick: To enforce rank 2, take the single-value decomposition $U\Sigma V^T$ of $F$ and remove the smallest eigenvalue of $\Sigma$. 
Hartley’s Normalized 8-point algorithm

Note that the entries in matrix $U$ vary by orders of magnitude:

\[
\begin{pmatrix}
10^6 & 10^6 & 10^3 & \cdots & 1
\end{pmatrix}
\begin{pmatrix}
(f_{11})
(f_{12})
(f_{13})
(f_{21})
(f_{22})
(f_{23})
(f_{31})
(f_{32})
(f_{33})
\end{pmatrix} = 0
\]

This causes numerical instability.

Trick: Rescale pixels so that mean squared difference is 2.

Horn’s Method -- “Relative Orientation” for Binocular Stereo Reconstruction: Compute R & t

Horn's Iterative Relative Orientation Method, 1990, computes R & t from corresponding rays. It does not use homogeneous coordinates (or F).

Also uses co-planarity of vectors $t, r_{\text{camera, left}}, r_{\text{camera, right}}$ to define an error function to minimize.

Uses a least squares approach to include $n$ matching 2D point pairs.

Uses a quaternion representation (we will see more about quaternions later).

Minimization is constrained by equations that express the physical properties of the problem (i.e., constraints on rotation matrix).

Resulting algorithm iteratively improves error (usually < 10 iterations needed).
Special Case Parallel Optical Axes: R & t given

left image

right image

Image Credit: Scharstein, 2014
Finding Matching Points: Follow Epipolar Lines & Template Match

Epipolar lines are parallel = along image rows (epipoles are at infinity)

Algorithm: Find corresponding points in same image rows via template matching (use normalized correlation coefficient to compute the match)
Result of Binocular Stereo Matching: Depth Map

\[ Z = \frac{bf}{\delta} \]

http://vision.middlebury.edu/stereo/data/scenes2014/
Parallel Optical Axes & Active Stereo with Structured Light

Active depth sensors that use IR: Kinect and iPhone, starting with iPhone X
Apple Face ID

Active stereo with structured light

projector

camera 1

camera 2

Image credit: Li Zhang
Active stereo with structured light

View without structured light

Image credit: Li Zhang
Active stereo with structured light

Project “structured” light patterns onto the object simplifies the correspondence problem

Image credit: Li Zhang
Active stereo with structured light

Image credit: Li Zhang
With the special case geometry – i.e., parallel optical axes, scene reconstruction is so much easier.

Why don’t we use it instead of the general case?
Rectification of Binocular Stereo Images: Undo Foreshortening

Why?
Epipolar lines are now parallel, enabling a simple search for corresponding points along image rows

Image Source:
Loop and Zhang, CVPR 1999
Rectification of Binocular Stereo Images: Undo Foreshortening

How?
Iterative Scheme

We want

\[ I_{\text{left}}(x + \delta/2, y) = I_{\text{right}}(x - \delta/2, y) \]

Least Squares Method:

\[
\min_{\delta} \sum_{p} \left[ I_{\text{left}}(x + \delta/2, y) - I_{\text{right}}(x - \delta/2, y) \right]^2
\]

\( p \) = patch
size of patch \( p \) = tradeoff
  too small  instability
  too large   smearing

Algorithm:
Use current estimate of disparity \( \delta \) to warp
Then solve LSM & update disparity

key image

offset image
Binocular Stereo Solution Paths: 2 Alternatives

1. “Weak Calibration”
   • If needed: Use rectification to ensure epipolar lines are along image rows
   • Find corresponding points in both views and calculate disparity $\delta$
   • Compute depth: $Z = bf/\delta$

2. “Strong Calibration”
   • Calibrate each camera (= interior orientation): $f$, $pp$
   • Find geometric transformation of cameras (= relative orientation): $R$, $t$
   • Find 3D coordinates
Binocular Stereo Solution Paths: 2 Alternatives

1. “Weak Calibration”
   • If needed: Use rectification to ensure epipolar lines are along image rows
   • Find corresponding points in both views and calculate disparity $\delta$
   • Compute depth: $Z = bf/\delta$

2. “Strong Calibration”
   • Calibrate each camera (= interior orientation): $f, pp$
   • Find geometric transformation of cameras (= relative orientation): $R, r_0$
   • Find 3D coordinates via triangulation

In our animal tracking research, “strong calibration” was the better solution
Binocular Stereo Solution Path: “Strong Calibration”

Throw wand in the air several times (mark out bird flying space)

Identify wand position in all views
Take advantage of knowing the dimensions of the wand

Estimate R and $r_0$

Images & Method: Theriault et al. 2014
Binocular Stereo Solution Path: “Strong Calibration”
Binocular Stereo for 3D Bird Flight Analysis

Images & Method:
Theriault et al. 2014
Calibration tool for thermal infrared cameras & Large Observation Spaces

Calibration tool with heat and ice packs

Images & Method: Theriault et al. 2014

CS 585: Image and Video Computing

© Betke
Calibration Apparatus and Reconstructed Points
Binocular Stereo Solution Path: “Strong Calibration”

Indoor scenario is much easier:

Instead of wand, use “checker board” as calibration device

Take many images at different positions & orientations

Image Source: Jean-Yves Bouguet
Binocular Stereo Solution Path: “Strong Calibration”

Indoor scenario is much easier:

Instead of wand, use “checker board” as calibration device

Take many images at different positions & orientations

Use
https://data.caltech.edu/records/jx9cx-fdh55
Or OpenCV

Image Source: Jean-Yves Bouguet
Code from my Research Lab:

Written by Diane Theriault

Published in Theriault et al., J Exp Biology, 2014

Figure S1: Software packages for easyCamera, easyWand, and easySBA and documentation can be downloaded from the OpenBU repository at http://hdl.handle.net/2144/8456. The Python SBA source code is also available at https://bitbucket.org/devangel77b/python-sba and the Python PIP stable release at https://pypi.python.org/pypi/sba/1.6.0
Reconstruction uncertainty due to quantization effects is shown for six hypothetical camera configurations. The cameras were simulated to have a pixel width of 18 \( \mu \text{m} \) and a field-of-view angle of 40.5 deg, and be positioned at a fixed height \( Z \) and aimed at a common, equidistant fixation point \( F=(0,0,Z) \). Horizontal cuts of the 3D view frustums of the cameras at height \( Z \) and lines at \( D_{\text{max}}=20 \) are shown from above.

Placing the cameras further apart reduces reconstruction uncertainty (A versus B).

If the cameras are placed too far apart (C), however, the view volume is ‘closed’, and there are unobservable regions of space where the cameras will be looking past each other.

If the distance between the outermost cameras is held constant, adding additional cameras may not decrease the uncertainty due to image quantization in the common observable region (D versus E).

If the image planes of the cameras are parallel (F), the common view volume is smaller and further away from the cameras than in the other configurations.

These 2D cuts of the 3D view frustums are at the level and elevation angle of the cameras; cuts at a different level or angle would show slightly greater reconstruction uncertainty but similar trends.
What is the impact on 3D reconstruction if the location detector is inaccurate?

Can the impact be quantified?

Field biologists really like to know how accurate the 3D estimates are!
Reconstruction uncertainty due to quantization and resolution issues is shown. In a video frame obtained for a flight study (A), the automatically detected locations of the animals may not be at their centers (colored dots in B). When estimating reconstruction uncertainty (C,D), we include this effect by corrupting the image projections of simulated world points, generated throughout the whole space, with Gaussian noise where the standard deviation is one-sixth of the calculated apparent size of an animal at that location (circles in B). When estimating the reconstruction uncertainty, including image location ambiguity (D) increases the estimated uncertainty more than threefold over image quantization alone (C) (note the change in color scale).
Figure S3: 3D flight trajectories of 28 Brazilian Free-tailed Bats during a 1-s interval are shown in the context of the spatially-varying reconstruction uncertainty arising due to both image quantization and image localization ambiguity from an oblique view (A) and from the top (B). The tracks are shown from the point of view of the cameras (C) and from the side (D). The observation distance between cameras and bats was approximately 10 m (B, D), chosen so that the nose-to-tail span of a bat in an image was at least 10 pixels. The baseline distance between the outermost cameras was approximately 6 m, chosen so that the expected uncertainty in reconstructed 3D positions at the observation distance due to image quantization and image localization ambiguity was less than 10 cm, the length of a bat. The RMS reconstruction uncertainty for the 1,656 estimated 3D positions shown was 7.8 cm.
Figure S4: The flight paths of 12 Cliff Swallows during a 2.3-s interval are shown in the context of the spatially-varying reconstruction uncertainty arising due to both image quantization and image localization ambiguity from an oblique view (A) and from the top (B). The tracks are shown from the point of view of the cameras (C) and from the side (D). At an observation distance of approximately 20 m (B,D), the birds, which are approximately 13 cm long, were imaged at an average length of 18 pixels. The baseline distance between the outermost cameras was approximately 11 m. The RMS reconstruction uncertainty for the 2,796 estimated 3D points shown was 5.9 cm, less than half the length of a bird.
First paper on Multiview Stereo and Using Internet Photo Collections to Reconstruct Scenes

**Goesele, Snavely, Curless, Hoppe, Seitz, ICCV 2007**

129 Flickr images taken by 98 photographers

merged model of Venus de Milo
First paper on Multiview Stereo and Using Internet Photo Collections to Reconstruct Scenes


56 Flickr images taken by 8 photographers

merged model of Pisa Cathedral
**Bundle Adjustment**

1950’s photogrammetry technique

Name: Bundles of light rays, originating from 3D points, used to adjust estimates

Goal: Solve simultaneously for 3D scene reconstruction and intrinsic & extrinsic parameters of each camera

Technique: Non-linear least squares method (use a package, e.g., ceres-solver.org)

Cost function to minimize: Reprojection error between the image locations of observed and predicted image points

\[
\min_{i \in \text{Cameras}} \sum_{j \in \text{Points}} \| r^{(j)}_{\text{camera},i} - \pi_i(r^{(j)}_{\text{camera},i}) \|^2
\]

where \( \pi_i \) is the mapping from an estimated 3D point into \( i \)th camera view
Bundle Adjustment is used to solve Structure-from-Motion Problems

Structure-from-Motion Problem:
Find 3D scene coordinates (here called “structure”) from a moving camera
Camera is usually calibrated (i.e., we have intrinsic parameters $f$ and $pp$)
Motion of camera yields a video where each frame has transformation parameters $R$ & $t$ that need to be estimated
Iterative Bundle Adjustment Algorithm:

**Input:** Images of scene or object taken by different cameras from different viewpoints

**Preprocessing:**
1. Extract features
2. Match corresponding features
3. Compute "scene graph" (definition: nodes=images, edges=camera transformation is plausible)
4. Initialize reconstruction based on 2 cameras in dense part of scene graph

**Repeat:**
1. Register a new image robustly to current 3D reconstruction
2. Add newly triangulated 3D points to current 3D reconstruction
3. Apply Bundle Adjustment to update current 3D reconstruction and camera parameters

**Output:** 3D reconstruction of scene or object
Rome dataset
74,394 images
What has changed since Deep Learning?

By and large, we still rely on conventional Bundle Adjustment to solve multi-view geometry for us.

While relatively reliable, this has major downsides:
Not online, not robust to scene motion, not amenable to end-to-end learning…

IMO we’re missing the correct way to “learn” multi-view geometry in a self-supervised way. It should be possible: Build a model that watches video and learns to reconstruct both pose and a proper 3D scene representation!

Maybe one of you will get there :)
Deep Learning Attempts at 3D Reconstruction

• Unsupervised Learning of Depth and Ego-Motion from Video, Zhou et al., CVPR 2017
• Deep Fundamental Matrix Estimation without Correspondences, Poursaeed et al., 2018
• BARF: Bundle-Adjusting Neural Radiance Fields, Lin et al., ICCV 2021
• The 8-Point Algorithm as an Inductive Bias for Relative Pose Prediction by ViTs, Rockwell et al., 2022
• Input-level Inductive Biases for 3D Reconstruction, Yifan et al., CVPR 2022
Parallel Tracking and Mapping for Small AR Workspaces

Klein & Murray, ISMAR 2007

https://www.youtube.com/watch?v=Y9HMn6bd-v8

Uses bundle adjustment
The Fundamental Matrix Song, Daniel Wedge:
Learning Objectives
You should be able to explain:

• Camera transformation problems
• Different representations of rotation
• Multiple measurement pairs (corresponding pixels in left & right cameras) are needed to reconstruct 3D coordinates of scene points
• Triangulation
• Epipolar geometry

• Projective geometry derivation of the fundamental matrix F
• Methods to compute F, R & t
• Special case of parallel optical axes
• Active stereo
• Weak & strong calibration
• Structure from motion
• Iterative Bundle Adjustment