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

BOSTON UNIVERSITY

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

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: $R\, \mathbf{r}_{camera} + \mathbf{t} = \mathbf{r}_{world}$

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: $R\, \mathbf{r}_{left} + \mathbf{r}_0 = \mathbf{r}_{right}$

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?

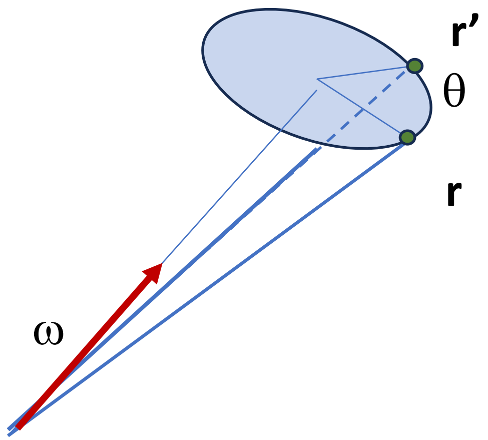
- Euler angles: roll, yaw, pitch (3 degrees of freedom)
- Quaternions *(more later, to prepare, review imaginary numbers)*
- Axis and angle: Axis is a unit vector ω (2 degrees of freedom)

  Angle: θ

Rodriguez' Formula:

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

$$\boldsymbol{\omega} \times \mathbf{r} = \begin{pmatrix} \omega_2 r_3 - \omega_3 r_2 \\ \omega_3 r_1 - \omega_1 r_3 \\ \omega_1 r_2 - \omega_2 r_1 \end{pmatrix}$$



Most popular representation: Rotation Matrix

Derivation of Rotation Matrix:

Rotation in the image plane by angle θ :

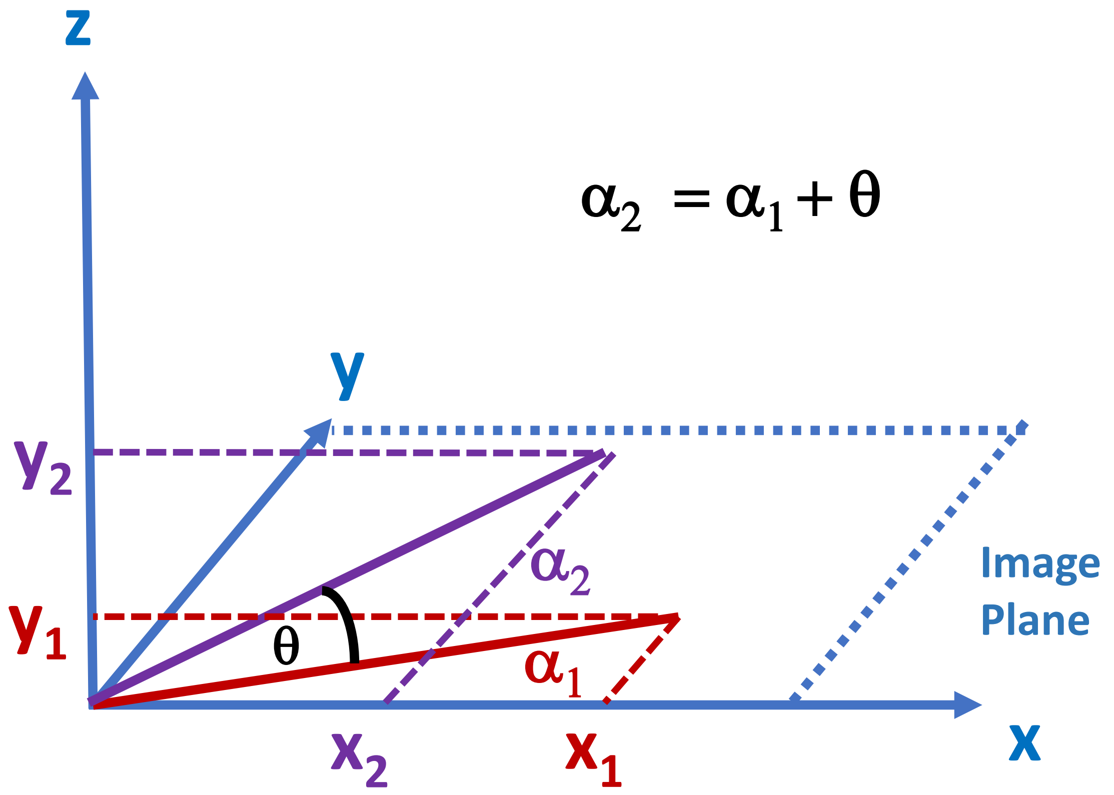
Original Position:

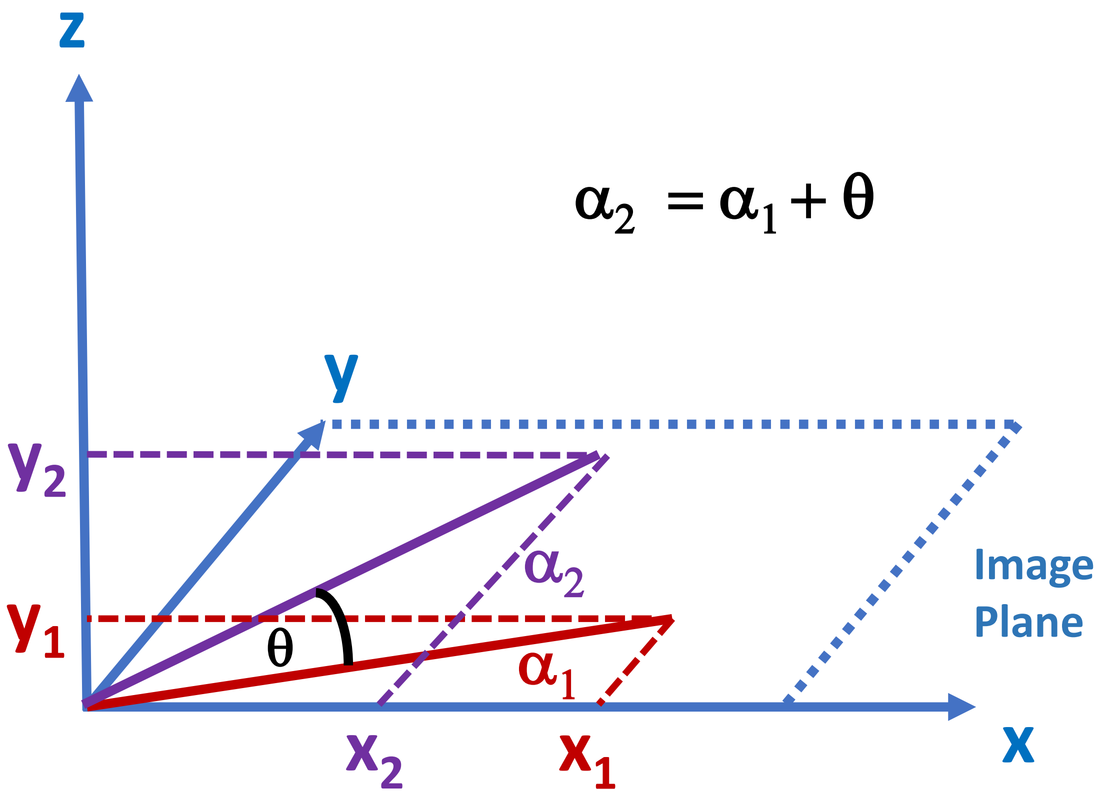
$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} r\cos\alpha_1 \\ r\sin\alpha_1 \\ 1 \end{pmatrix}$$

Rotated Position:

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

$$\alpha_2 = \alpha_1 + \theta$$





$$\alpha_2 = \alpha_1 + \theta$$

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

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

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

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

$R^TR = RR^T = I$

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: $R\ \mathbf{r}_{camera} + \mathbf{t} = \mathbf{r}_{world}$

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

3. Absolute Orientation = Alignment of 2 Cameras or 2 Medical Scans

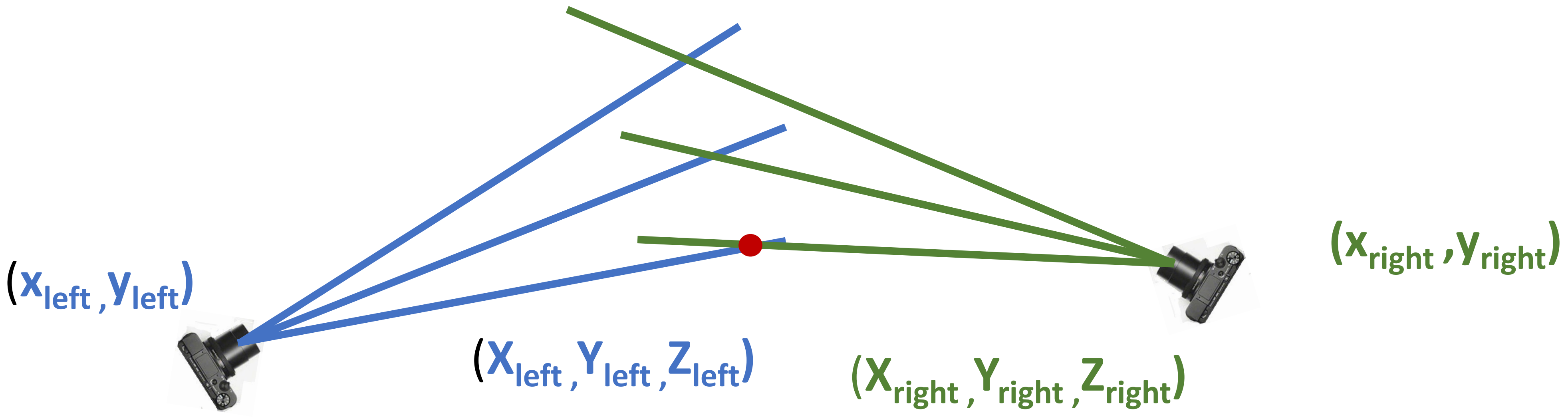
   Find relationship between cameras. 3D coordinates of points are known

4. Relative Orientation = Alignment of 2 Cameras

   Find relationship between cameras. 3D coordinates not known, only rays known

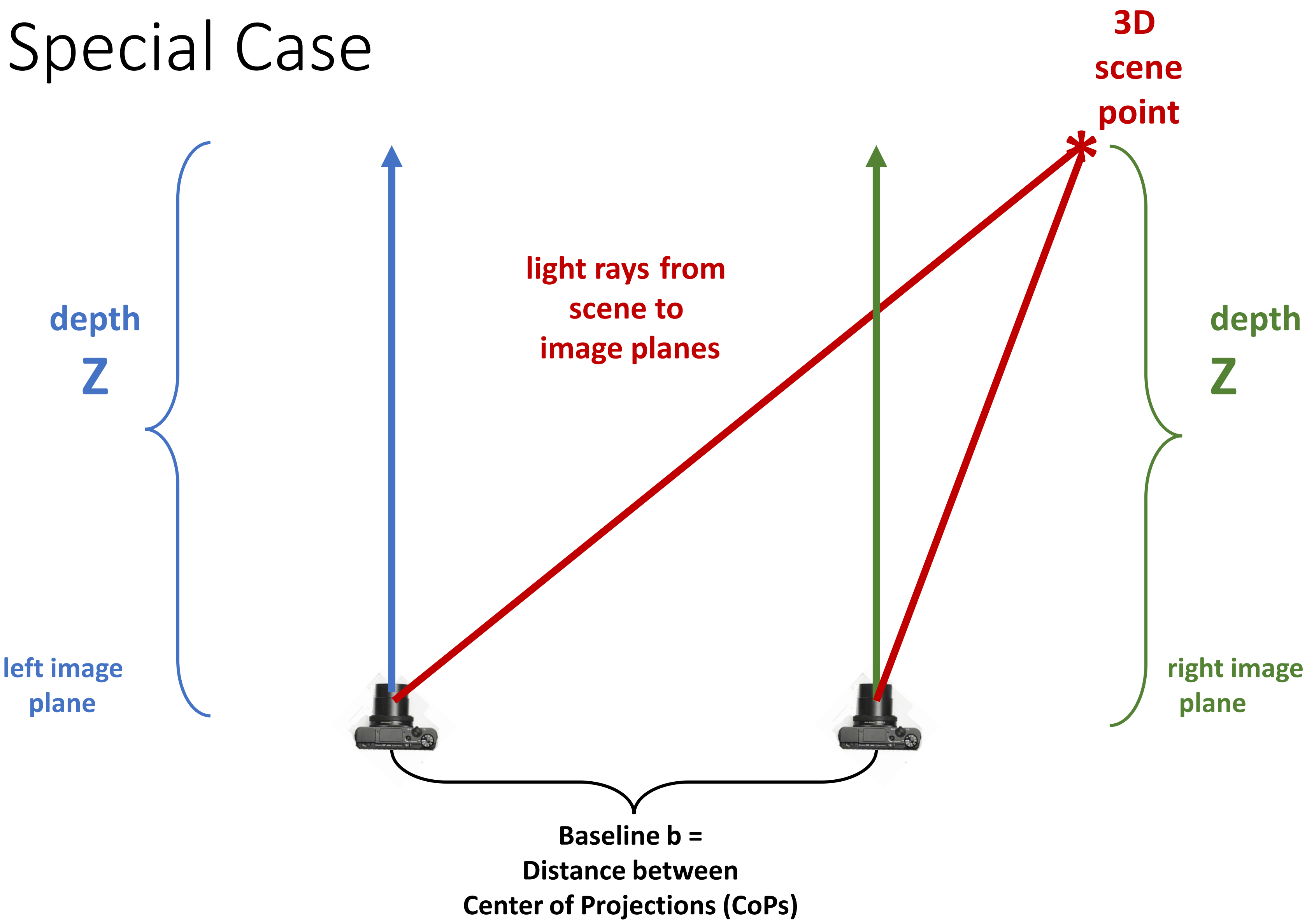
Relative Orientation for Binocular Stereo

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



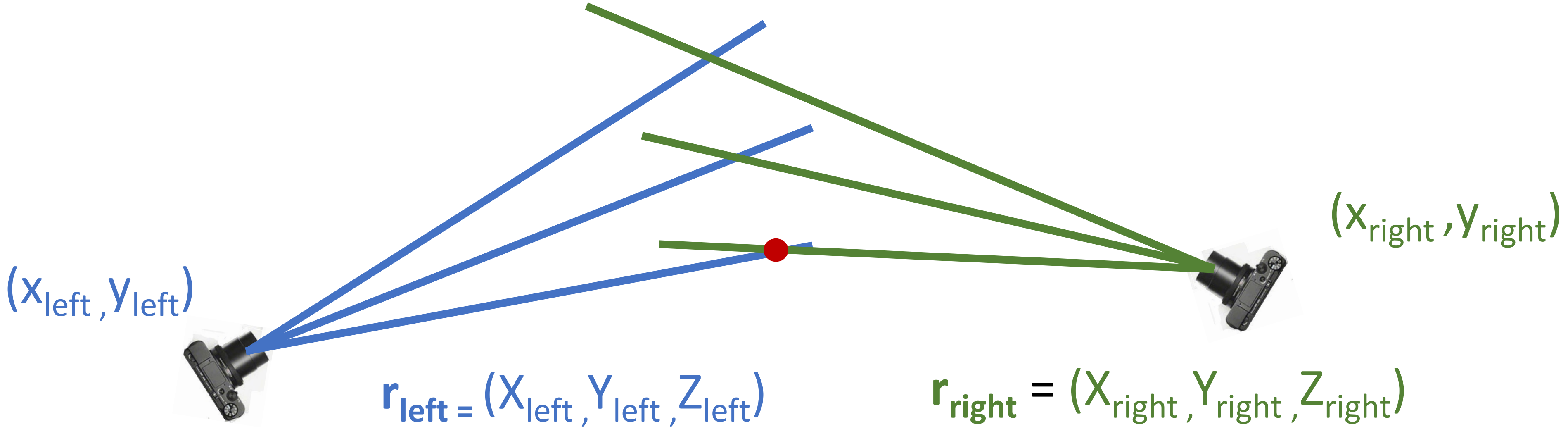
Given: 2D coordinates of image points of same world point

Special Case



$$Z = bf/\delta$$

Relative Orientation = General Binocular Stereo



Use perspective projection equations:

$x_{left}/f_{left} = X_{left}/Z_{left}$ $\qquad$ $y_{left}/f_{left} = Y_{left}/Z_{left}$

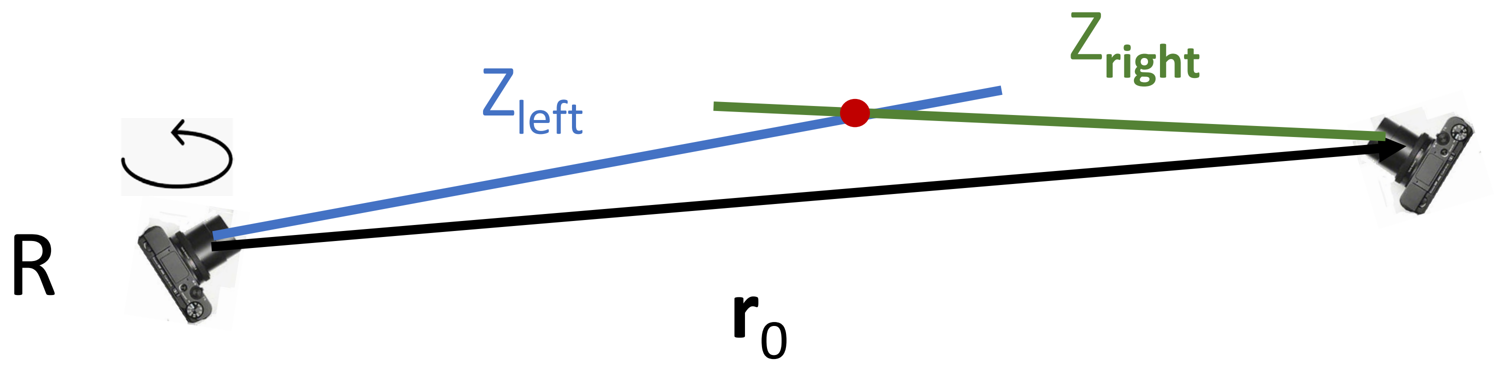
$x_{right}/f_{right} = X_{right}/Z_{right}$ $\qquad$ $y_{right}/f_{right} = Y_{right}/Z_{right}$

Transformation equation: $R\,\mathbf{r}_{left} + \mathbf{r}_0 = \mathbf{r}_{right}$ $\qquad$ R = rotation matrix, $\mathbf{r}_0$ = translation

Relative Orientation for Binocular Stereo

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

Unknown: Rotation matrix R, translation $\mathbf{r}_0$, Z coordinates of $\mathbf{r}_{left}$, $\mathbf{r}_{right}$

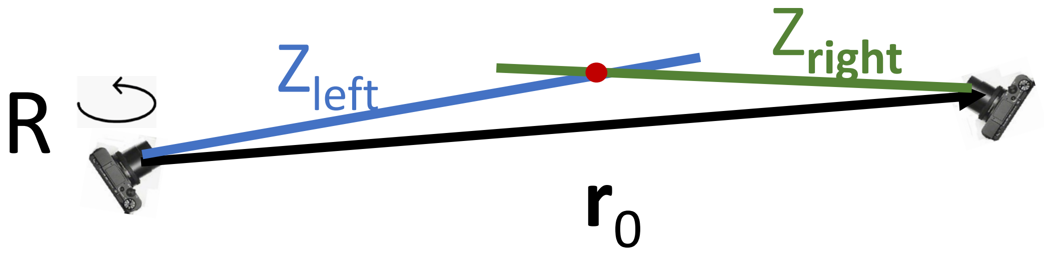


Relative Orientation for Binocular Stereo

$R\,\mathbf{r}_{left} + \mathbf{r}_0 = \mathbf{r}_{right}$

is equivalent to:

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



Insert Perspective Projection Equations:

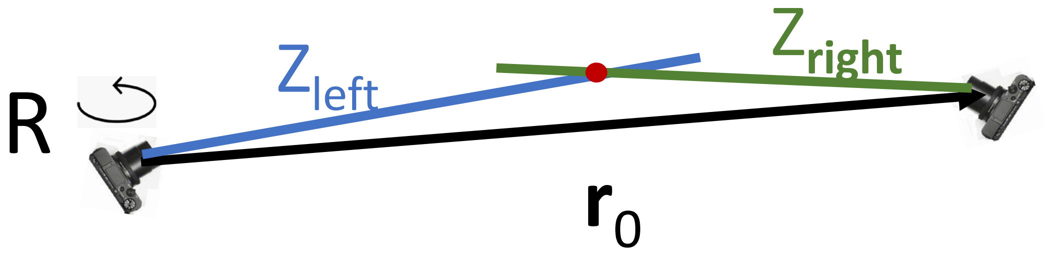
$x_{left}/f = X_{left}/Z_{left}$ $\quad y_{left}/f = Y_{left}/Z_{left}$

$x_{right}/f = X_{right}/Z_{right}$ $\quad y_{right}/f = Y_{right}/Z_{right}$

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

Multiply by f/Z_{left}

Relative Orientation for Binocular Stereo



$R\,\mathbf{r}_{left} + \mathbf{r}_0 = \mathbf{r}_{right}$

is equivalent to:

$$r_{11}\, x_{left} + r_{12}\, y_{left} + r_{13}\, f + r_{14}\, f/ Z_{left} = x_{right}\, Z_{right}/Z_{left}$$

$$r_{21}\, x_{left} + r_{22}\, y_{left} + r_{23}\, f + r_{24}\, f/Z_{left} = y_{right}\, Z_{right}/Z_{left}$$

$$r_{31}\, x_{left} + r_{32}\, y_{left} + r_{33}\, f + r_{34}\, f/Z_{left} = f\, Z_{right}/Z_{left}$$

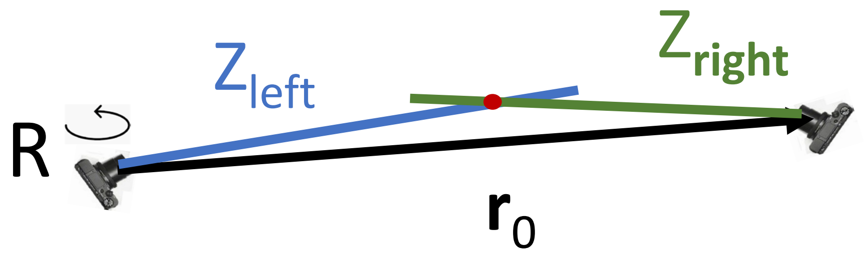
One measurement pair (x_{left}, y_{left}) and (x_{right}, y_{right}) => 3 equations

with 14 unknowns r_{11}, r_{12}, …, r_{34}, and Z_{right}, Z_{left}

Relative Orientation

$R\ \mathbf{r}_{left} + \mathbf{r}_0 = \mathbf{r}_{right}$

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



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

with 12 unknown $r_{11}, r_{12}, \ldots, r_{34}$ and 2 unknown Z_{right}, Z_{left}

Trick: To solve for 14 unknowns:

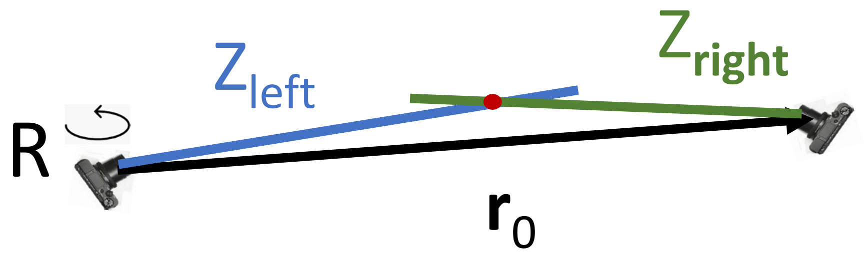
Use n measurements => 3n equations

Find additional equations

Relative Orientation

$R\,\mathbf{r}_{left} + \mathbf{r}_0 = \mathbf{r}_{right}$

$$
\begin{aligned}
r_{11}\, x_{left} + r_{12}\, y_{left} + r_{13}\, f + r_{14}\, f/Z_{left} &= x_{right}\, Z_{right}/Z_{left} \\
r_{21}\, x_{left} + r_{22}\, y_{left} + r_{23}\, f + r_{24}\, f/Z_{left} &= y_{right}\, Z_{right}/Z_{left} \\
r_{31}\, x_{left} + r_{32}\, y_{left} + r_{33}\, f + r_{34}\, f/Z_{left} &= f\, Z_{right}/Z_{left}
\end{aligned}
$$



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

with 12 unknown r_{11}, r_{12}, …, r_{34}, and 2 unknown Z_{right}, Z_{left}

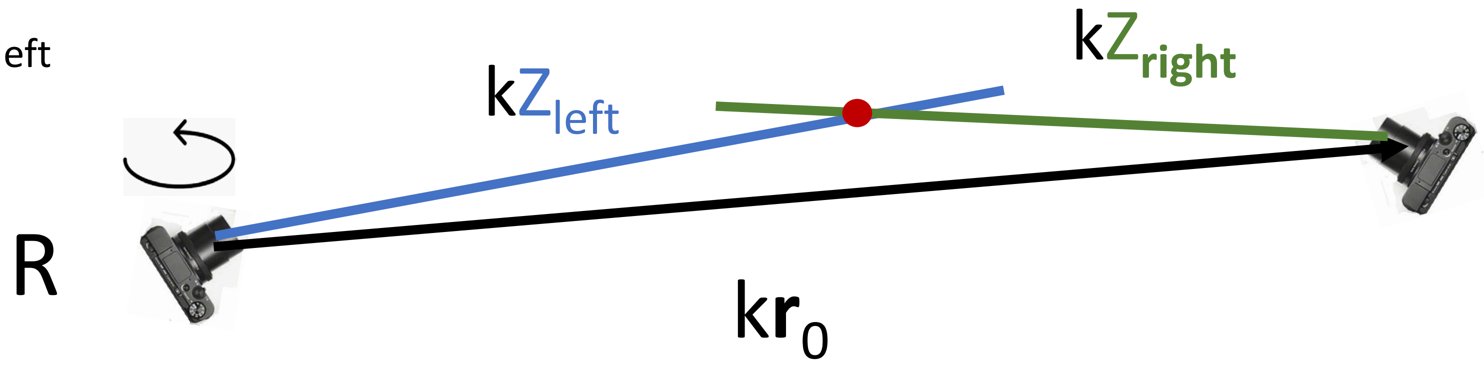
One extra equation:

Scale factor ambiguity $\mathbf{r}_0$, Z_{right}, Z_{left}

$\Leftrightarrow$ $k\mathbf{r}_0$, kZ_{right}, kZ_{left}

Force $\mathbf{r}_0$ to be unit vector

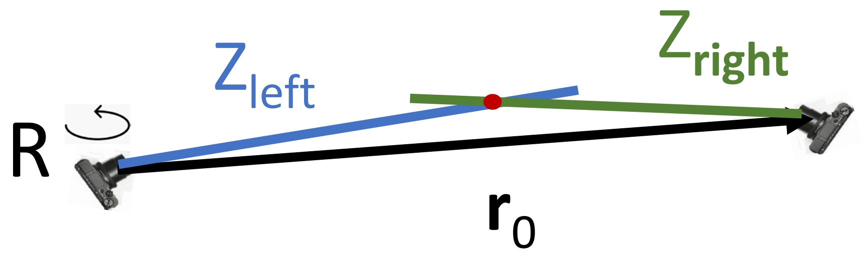
=> $|\mathbf{r}_0| = 1$



Relative Orientation

$R\,\mathbf{r}_{left} + \mathbf{r}_0 = \mathbf{r}_{right}$

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



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

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

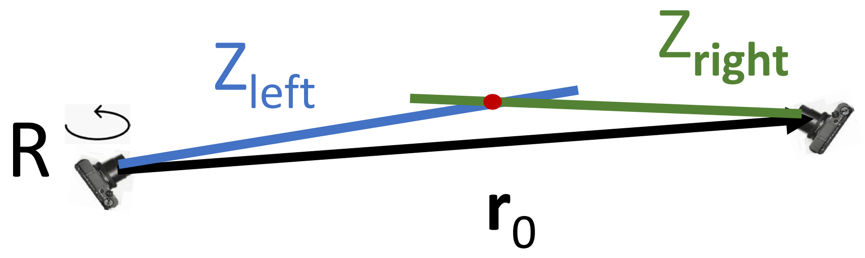
# unknowns:	12	for R, $\mathbf{r}_0$
	2n	for Z_{right}, Z_{left} for each of n pairs of measurements

12 + 2n unknowns

Relative Orientation

$R\,\mathbf{r}_{left} + \mathbf{r}_0 = \mathbf{r}_{right}$

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



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

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

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

1 for unit length translation: $|\mathbf{r}_0|=1$

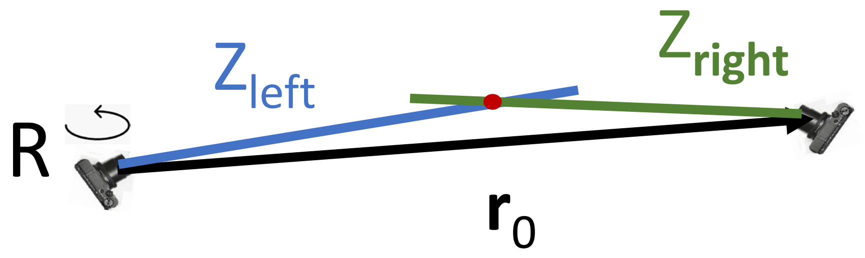
3n for 3 equations per measurement pair

7+ 3n equations

Relative Orientation

$\mathbf{R}\,\mathbf{r}_{left} + \mathbf{r}_0 = \mathbf{r}_{right}$

$$
\begin{aligned}
r_{11}\, x_{left} + r_{12}\, y_{left} + r_{13}\, f + r_{14}\, f/Z_{left} &= x_{right}\, Z_{right}/Z_{left} \\
r_{21}\, x_{left} + r_{22}\, y_{left} + r_{23}\, f + r_{24}\, f/Z_{left} &= y_{right}\, Z_{right}/Z_{left} \\
r_{31}\, x_{left} + r_{32}\, y_{left} + r_{33}\, f + r_{34}\, f/Z_{left} &= f\, Z_{right}/Z_{left}
\end{aligned}
$$



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

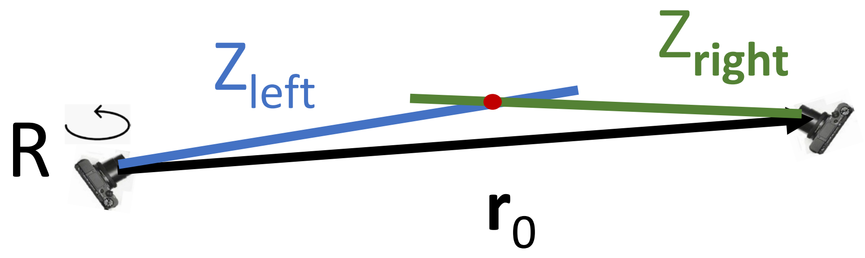
# unknowns:	12	for R, $\mathbf{r}_0$
	2n	for Z_{right}, Z_{left} for each of n pairs of measurements
Number of equations:	6	for orthonormal R (columns sum to 1, dot products 0)
	1	for unit length translation $\mathbf{r}_0$
	3n	for 3 equations per measurement pair

Need at least n = ? measurement pairs: $12 + 2 * n = 7 + 3*n$

Relative Orientation

$R\,\mathbf{r}_{left} + \mathbf{r}_0 = \mathbf{r}_{right}$

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



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

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

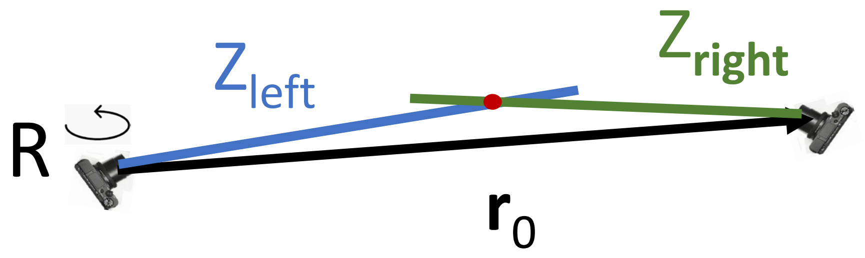
# unknowns:	12	for R, $\mathbf{r}_0$
	2n	for Z_{right}, Z_{left} for each of n pairs of measurements
Number of equations:	6	for orthonormal R (columns sum to 1, dot products 0)
	1	for unit length translation $\|\mathbf{r}_0\|=1$
	3n	for 3 equations per measurement pair

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

Relative Orientation

$R\,\mathbf{r}_{left} + \mathbf{r}_0 = \mathbf{r}_{right}$

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



n measurement pairs (x_{left}, y_{left}) and (x_{right}, y_{right}) => $3n$ equations + 7

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

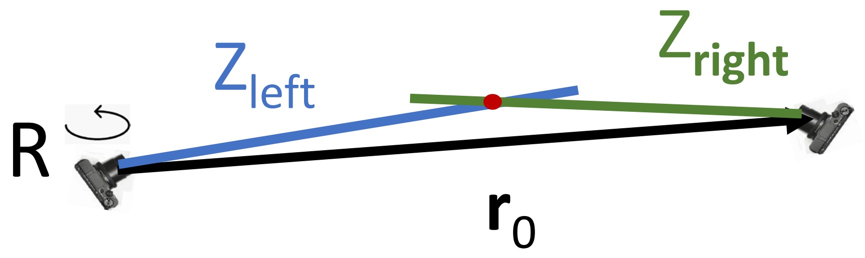
Need *at least* 5 measurement pairs --

Does that mean 5 pairs are enough?

Relative Orientation

$R\,\mathbf{r}_{left} + \mathbf{r}_0 = \mathbf{r}_{right}$

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



14 unknowns r_{11}, r_{12}, …, r_{34}, and Z_{right}, Z_{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…

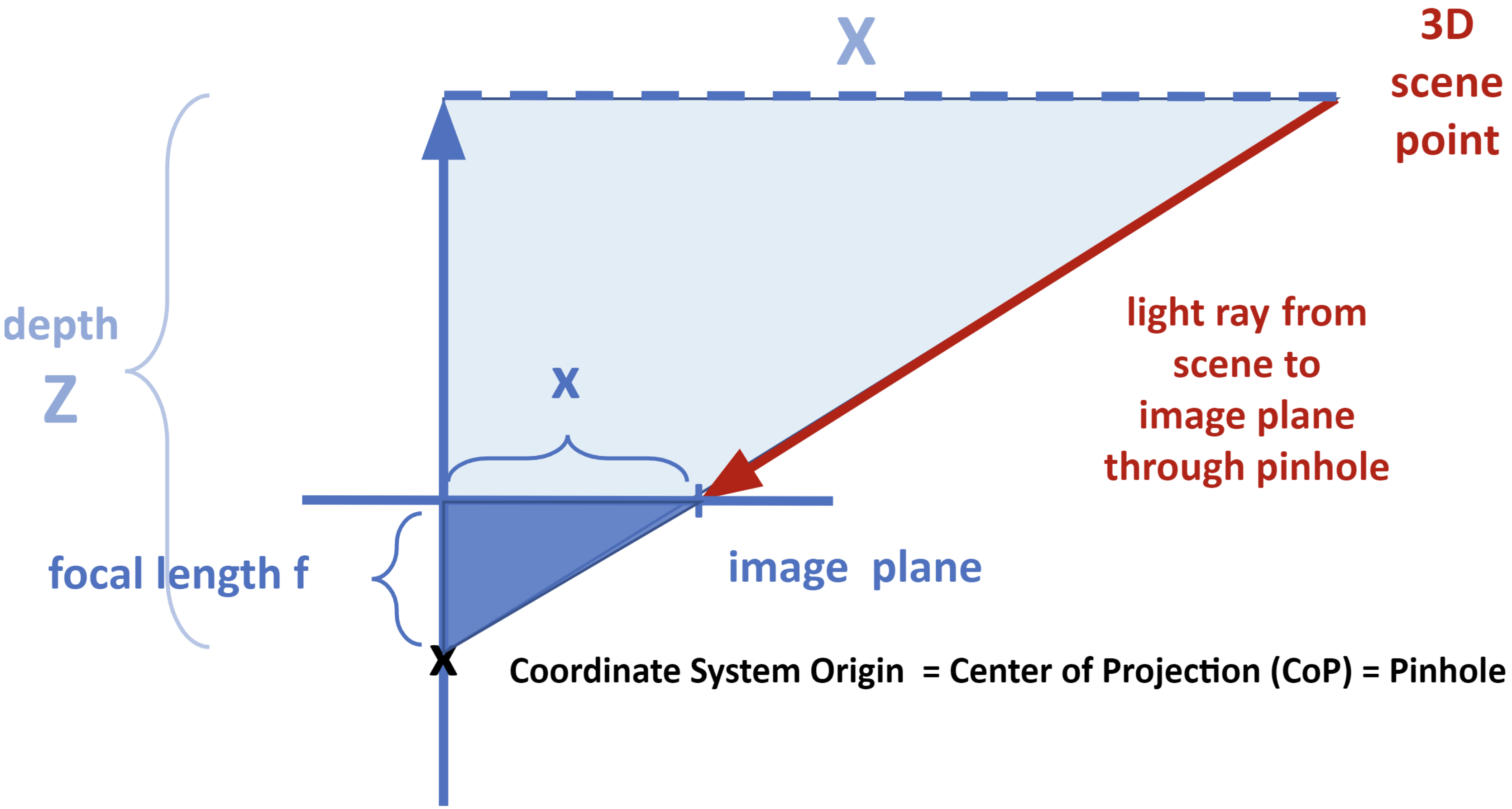
“Elegant” Computer Vision Math: Projective Geometry

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 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:

$X/Z = x/f$

$x = f\,X/Z$

$y = f\,Y/Z$

"Elegant" Computer Vision Math: Projective Geometry

Cartesian coordinates (="heterogeneous" coordinates):

Image point $(x,y)^T = (f\,X/Z,\ f\,Y/Z)^T$

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 $(p_x, p_y)^T$:

$$\begin{pmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \underbrace{\begin{pmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix}}_{K} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{[\,I \mid 0\,]} = K\,[\,I \mid 0\,]$$

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

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 $(p_x,p_y)^T$:

$$\begin{pmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} = \underbrace{\begin{pmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix}}_{K} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{[\, I \mid 0 \,]} = K\,[\, I \mid 0 \,]$$

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 90^0
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
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: $R\,\mathbf{r}_{camera} + \mathbf{t} = \mathbf{r}_{world}$

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: $R\, \mathbf{r}_{camera} + \mathbf{t} = \mathbf{r}_{world}$

$$R\, \mathbf{r}_{camera} = \mathbf{r}_{world} - \mathbf{t}$$

$$R^T R\, \mathbf{r}_{camera} = R^T (\mathbf{r}_{world} - \mathbf{t})$$

$$\mathbf{r}_{camera} = R^T (\mathbf{r}_{world} - \mathbf{t})$$

R here is $\mathbf{R}^T$

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: $R\,\mathbf{r}_{camera} + \mathbf{t} = \mathbf{r}_{world}$

$$R\,\mathbf{r}_{camera} = \mathbf{r}_{world} - \mathbf{t}$$

$$R^T R\,\mathbf{r}_{camera} = R^T(\mathbf{r}_{world} - \mathbf{t})$$

$$\mathbf{r}_{camera} = R^T(\mathbf{r}_{world} - \mathbf{t})$$

heterogenous (= 'regular') 3D vectors $\mathbf{r}_{camera}$, $\mathbf{r}_{world}$, $\mathbf{t}$

R here is $\mathbf{R}^T$

In homogeneous coordinates: $\tilde{\mathbf{r}}_{camera} = \underbrace{\begin{pmatrix} \mathbf{R}_{3x3} & -\mathbf{R}\tilde{\mathbf{t}} \\ \mathbf{0}_{1x3} & 1 \end{pmatrix}}_{C^{W2C}} \tilde{\mathbf{r}}_{world}$ or

$$\tilde{\mathbf{r}}_{world} = \underbrace{\begin{pmatrix} \mathbf{R}^T & \tilde{\mathbf{t}} \\ \mathbf{0} & 1 \end{pmatrix}}_{C^{C2W}} \tilde{\mathbf{r}}_{camera}$$

homogeneous 4D vectors $\tilde{\mathbf{r}}_{camera}$, $\tilde{\mathbf{r}}_{world}$, $\tilde{\mathbf{t}}$

Projective Geometry:
Mapping World Coordinates to Image Coordinates

Interior calibration — Interior calibration — exterior calibration

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

$$\tilde{\mathbf{r}}_{\mathbf{image}} = K\,[\,I \,|\, 0\,]\, C^{W2C}\, \tilde{\mathbf{r}}_{\mathbf{world}}$$

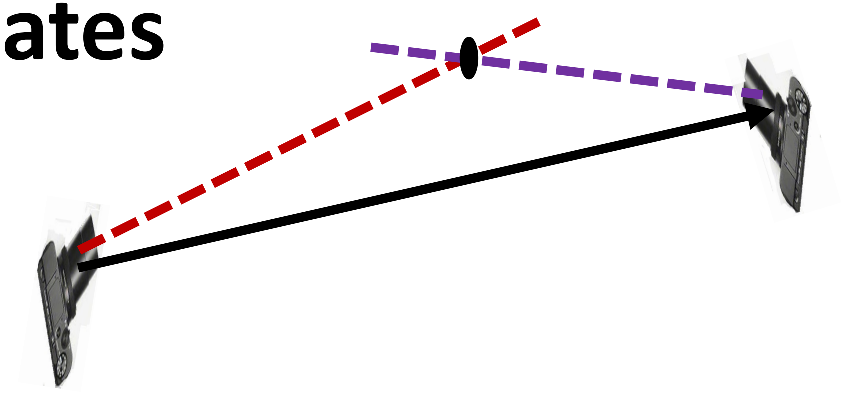
$$\boxed{\tilde{\mathbf{r}}_{\mathbf{image}} = \mathbf{P}\, \tilde{\mathbf{r}}_{\mathbf{world}}}$$

(also written as $\pi(\tilde{\mathbf{r}}_{\mathbf{world}})$ 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 $\mathbf{r}_{world}$ from a matching pair of image points $\mathbf{r}_{image1}$ & $\mathbf{r}_{image2}$

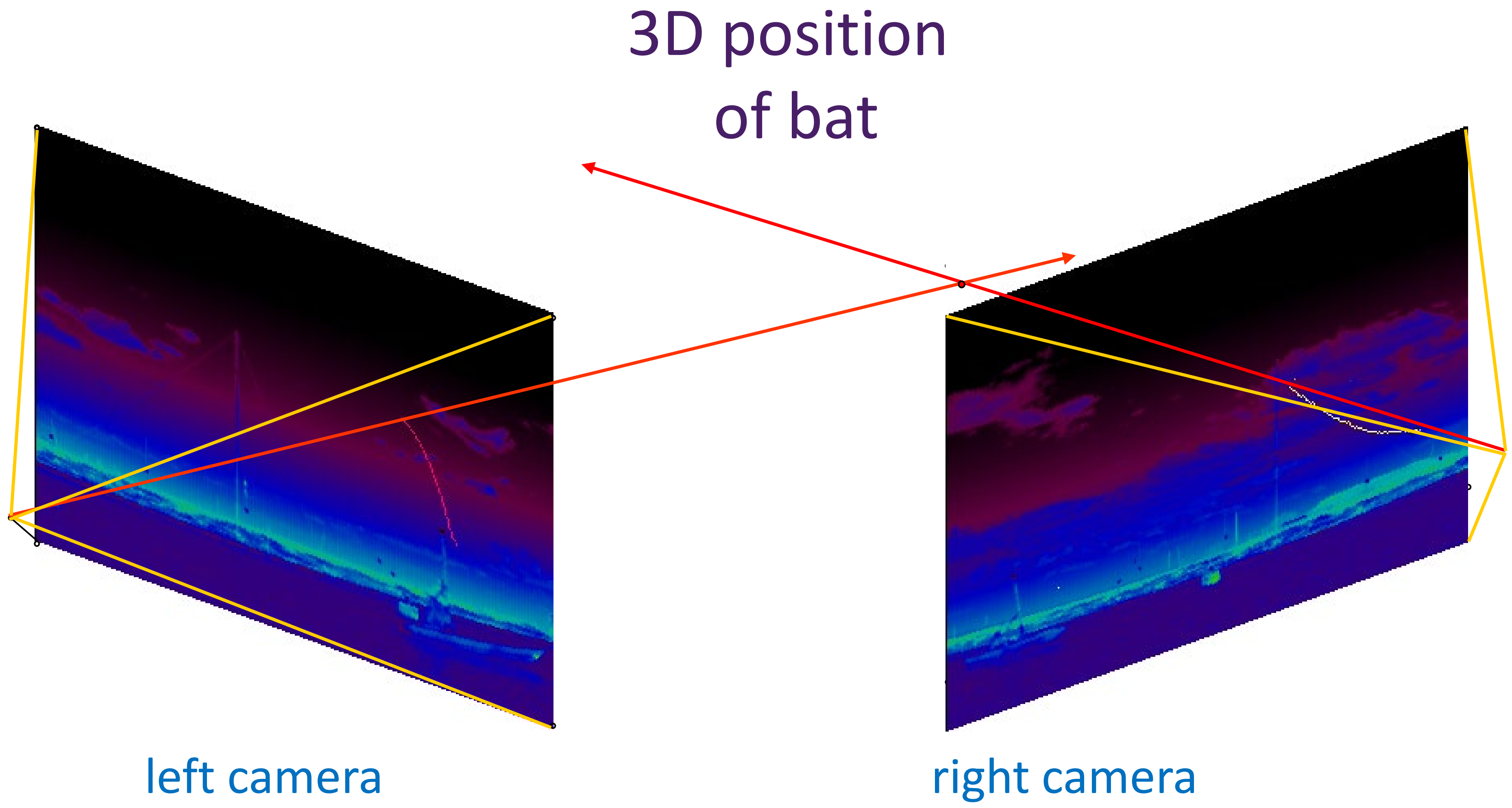
If no error: $|\pi_i(\mathbf{r}_{world}) - \mathbf{r}_{image,i}| = 0$

But likely errors, so use a least squares approach:

$$\mathbf{r}_{world,best} = \text{argmin} \sum |\pi_i(\mathbf{r}_{world}) - \mathbf{r}_{image,i}|^2$$

BOSTON UNIVERSITY

Stereoscopic 3D Reconstruction with Triangulation



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)

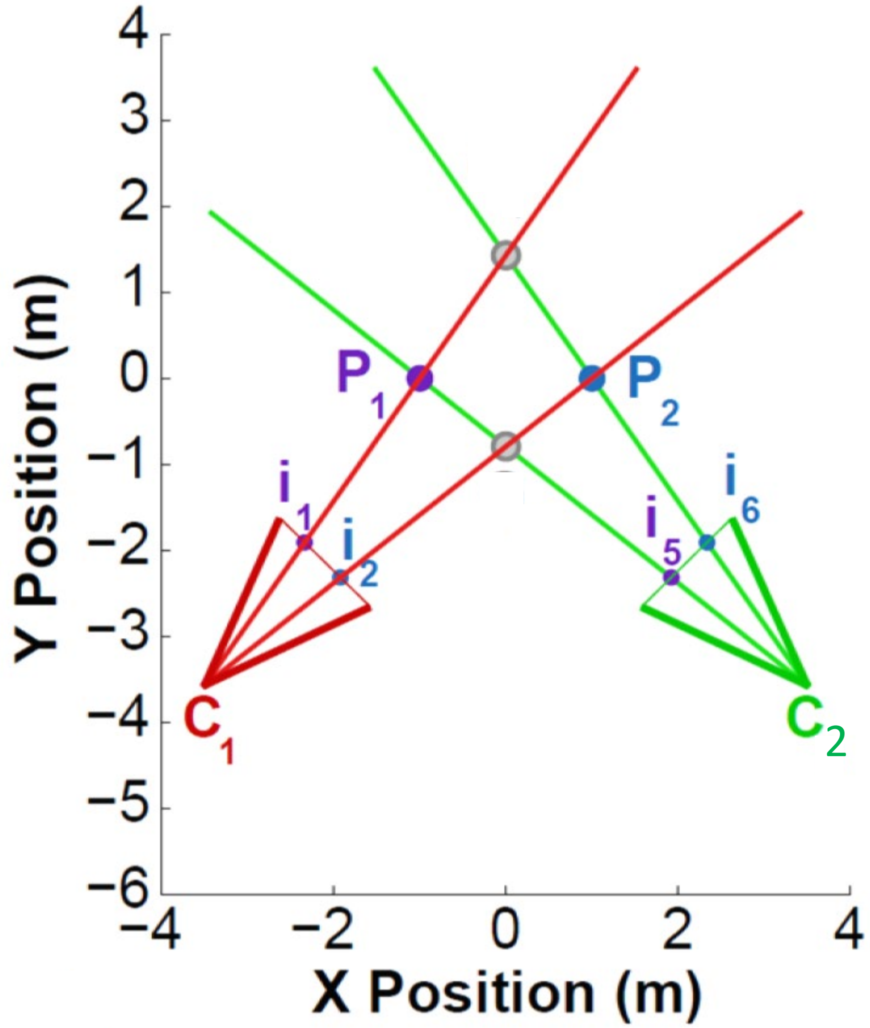
- Variations of the 8-point Algorithm

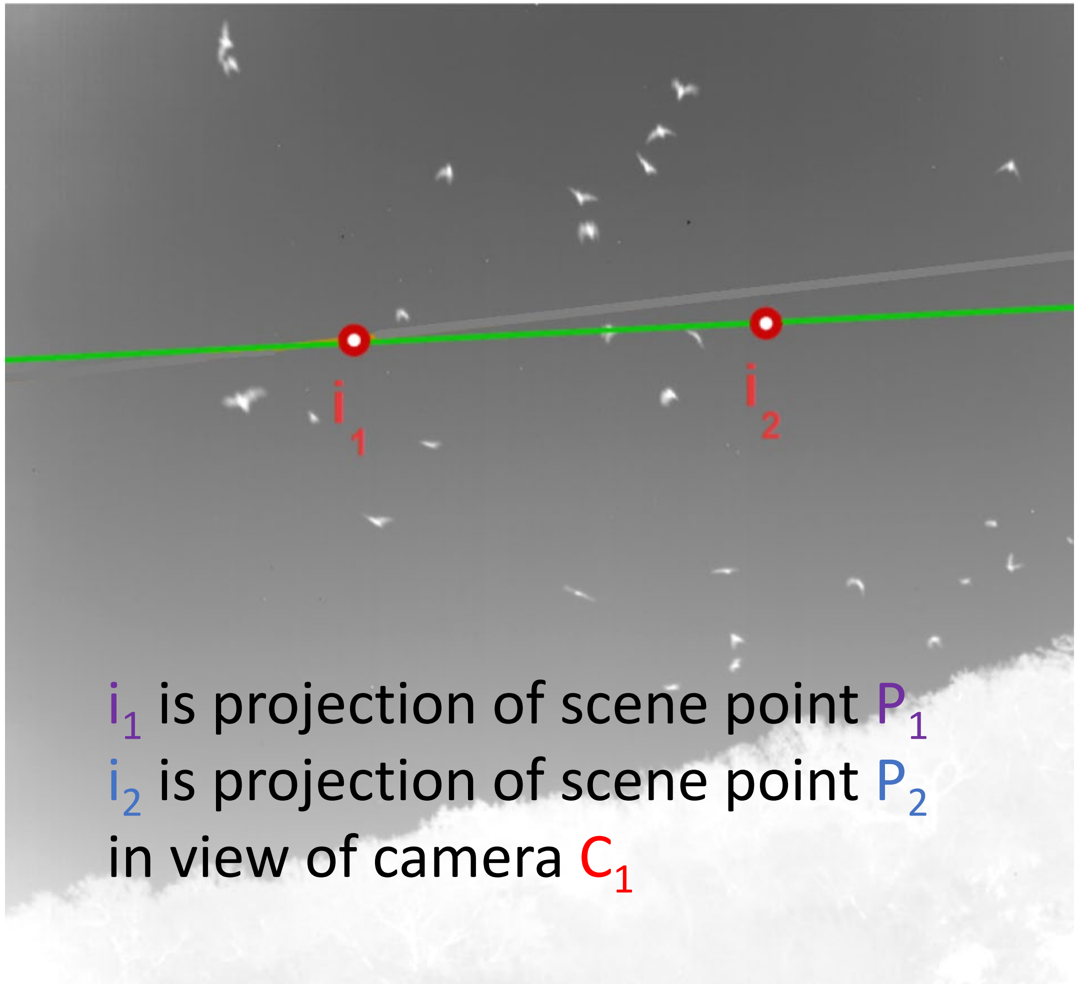
    e.g. Hartley's Normalized 8-point algorithm (1997)

- 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

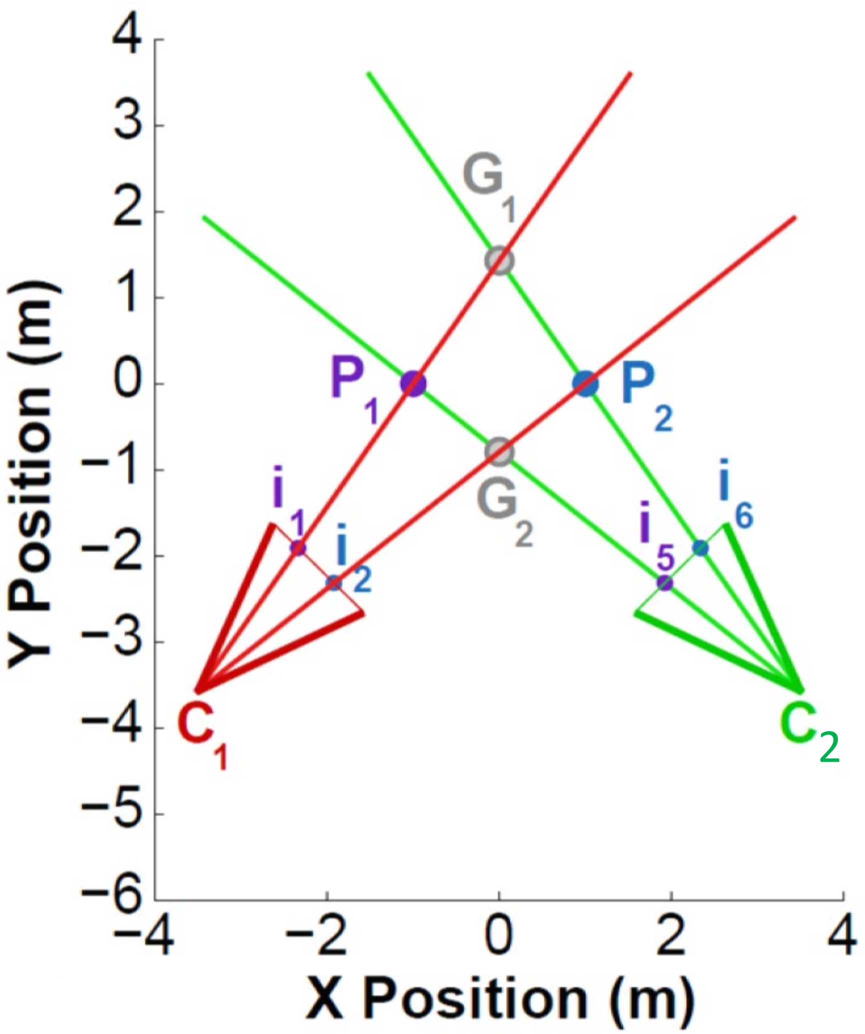
View from above: Z axis = direction of gravity

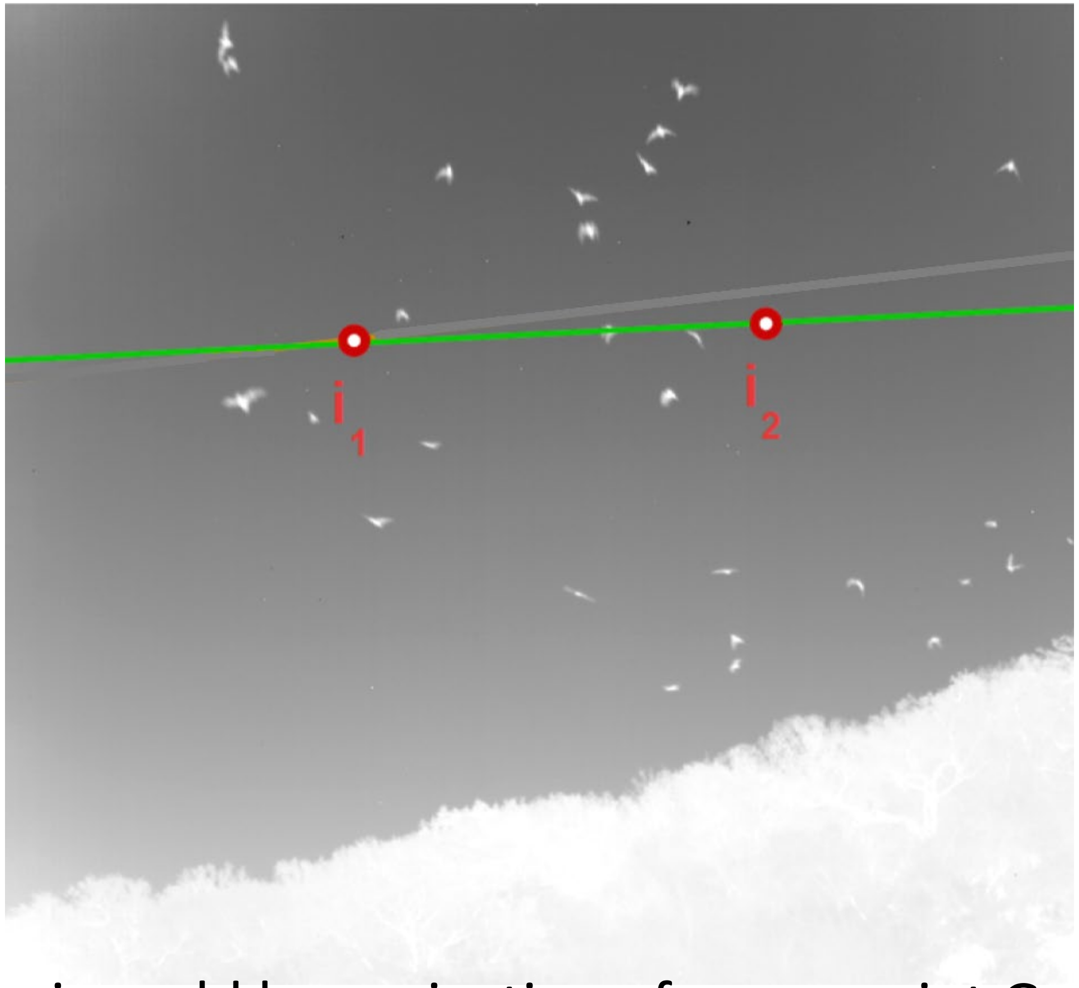




i_1 is projection of scene point P_1
i_2 is projection of scene point P_2
in view of camera C_1

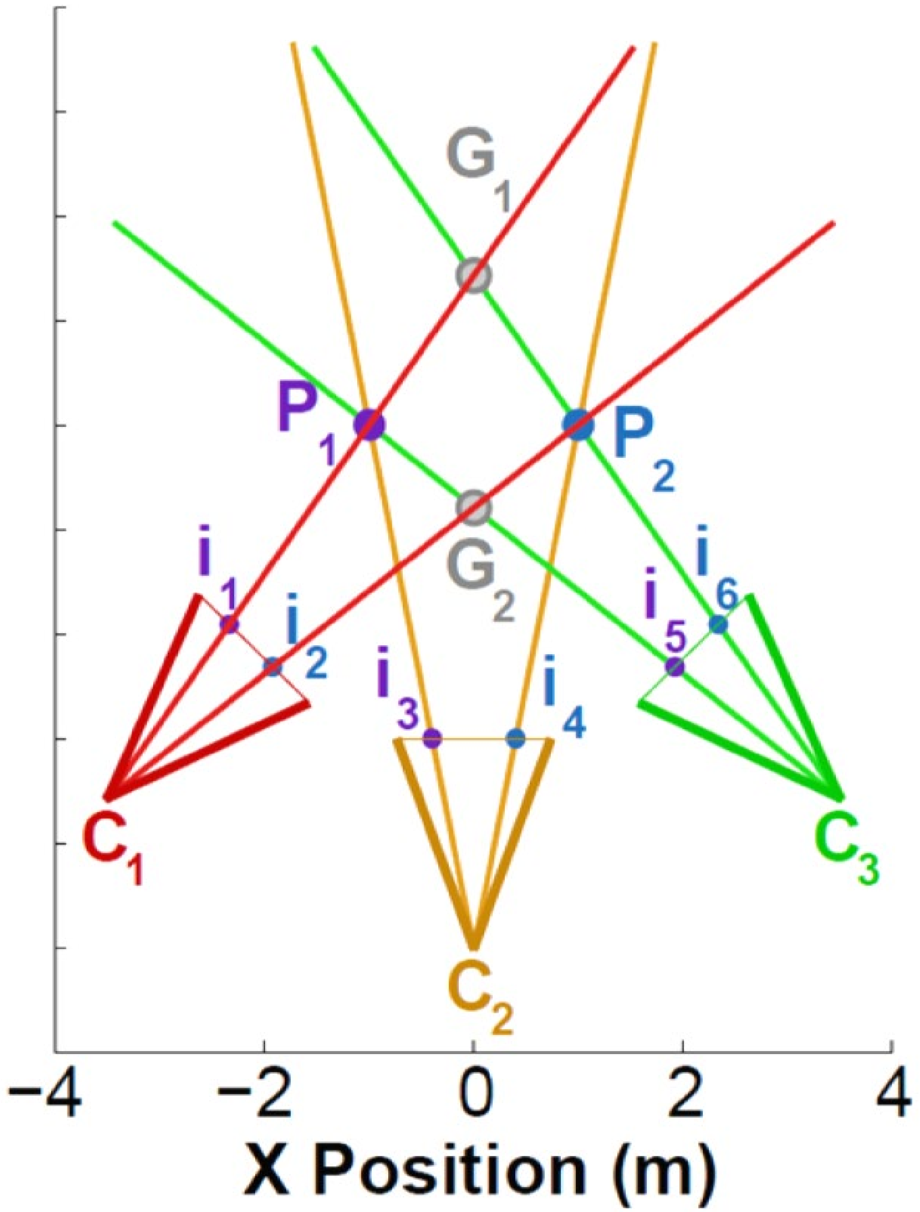
Multi-Camera Stereo

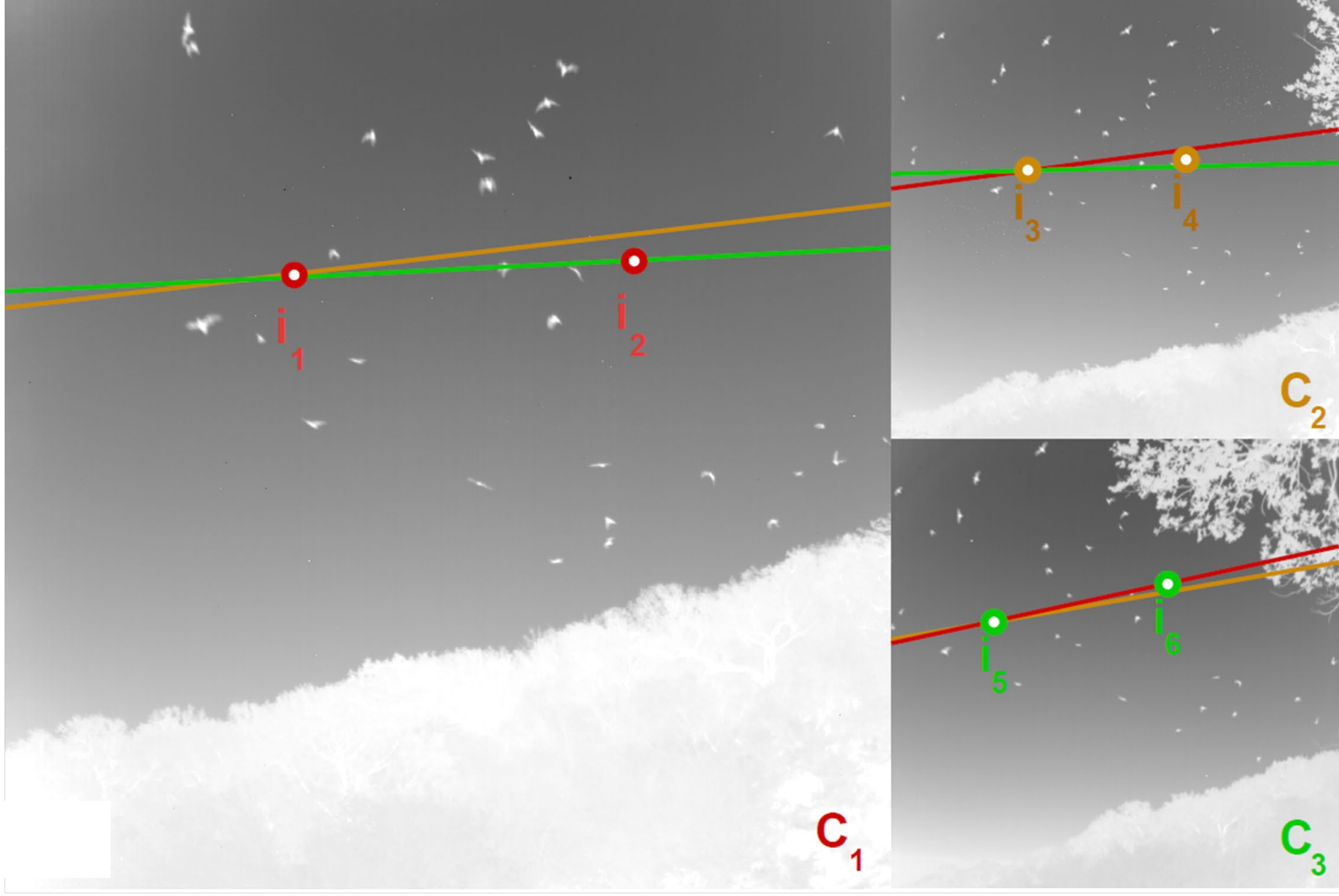




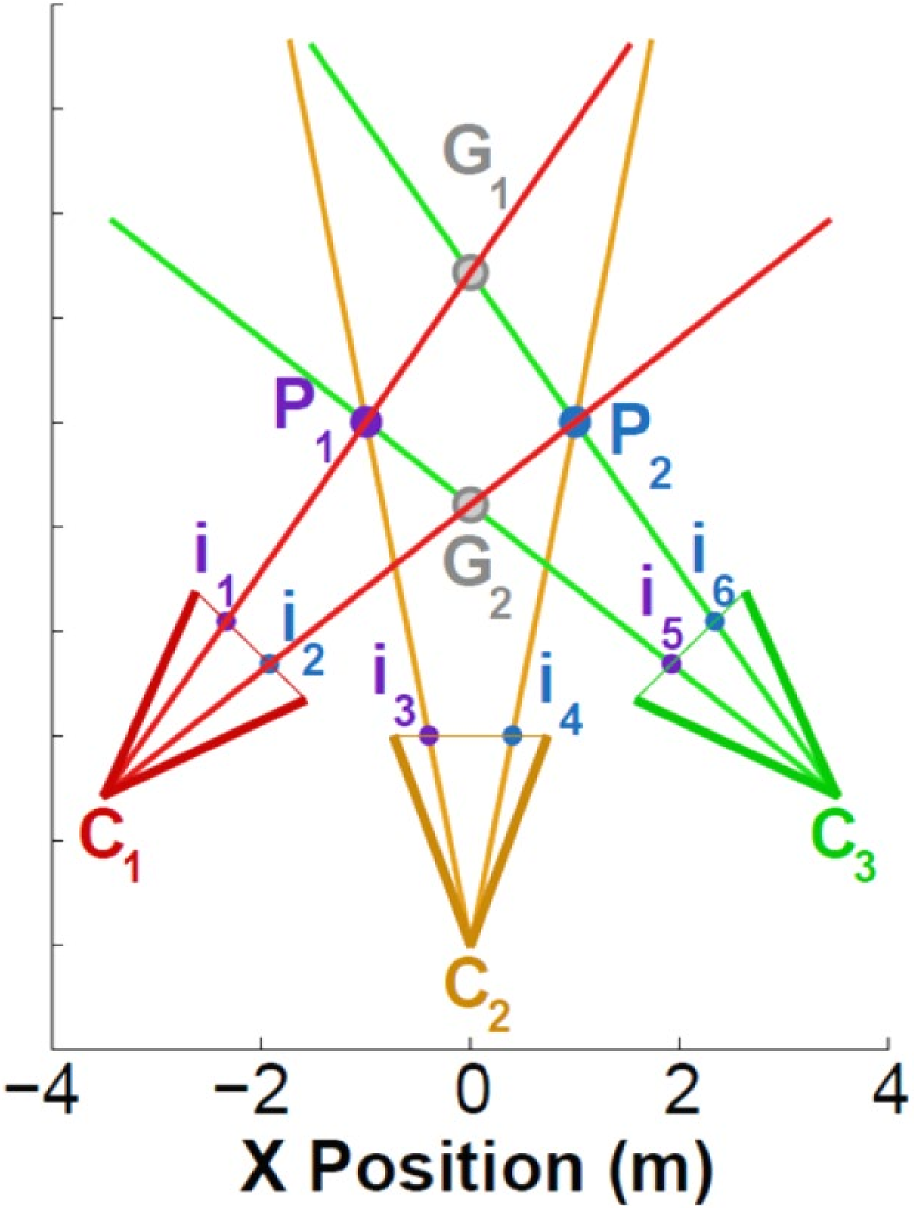
i_1 could be projection of scene point G_1

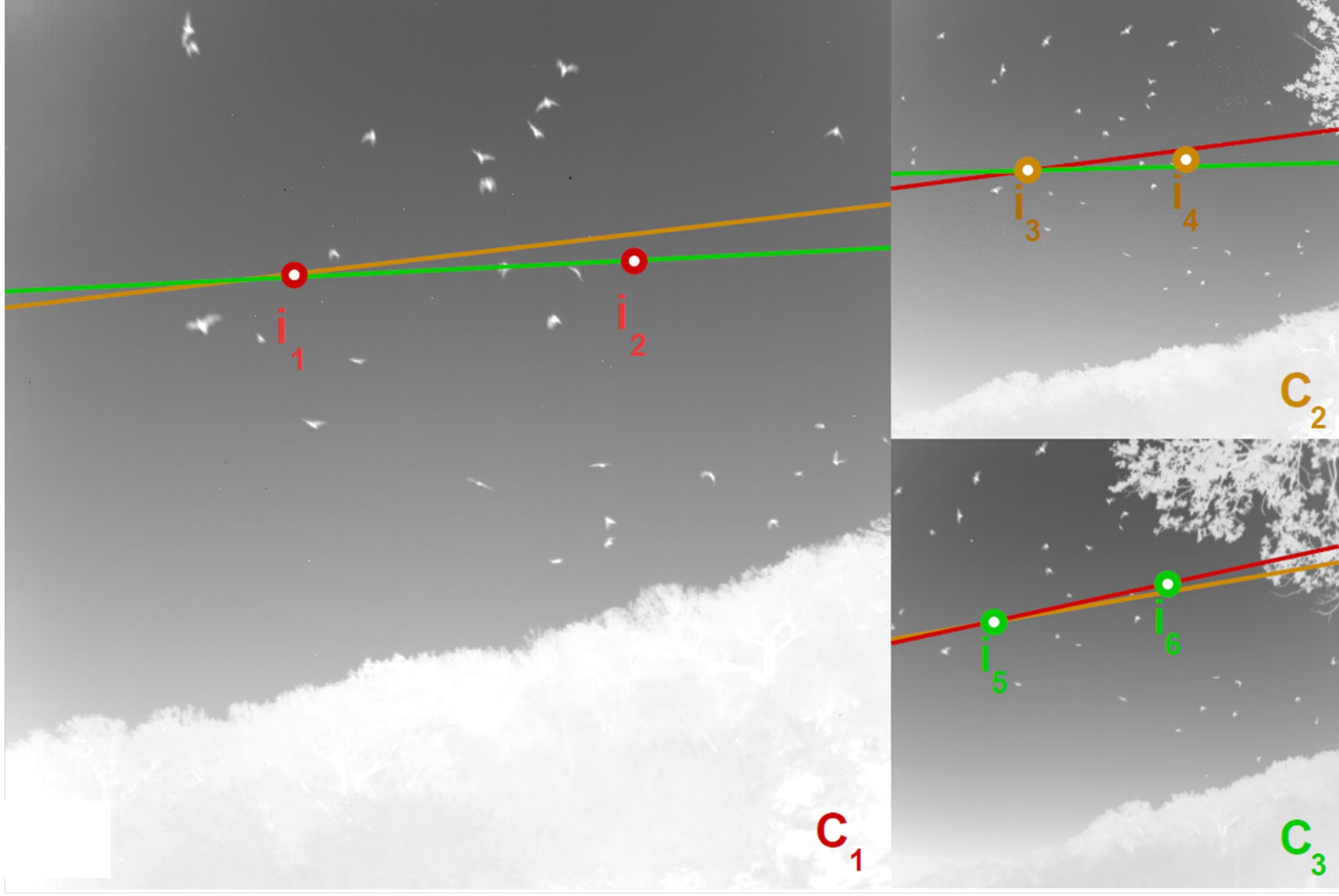
i_2 could projection of scene point G_2



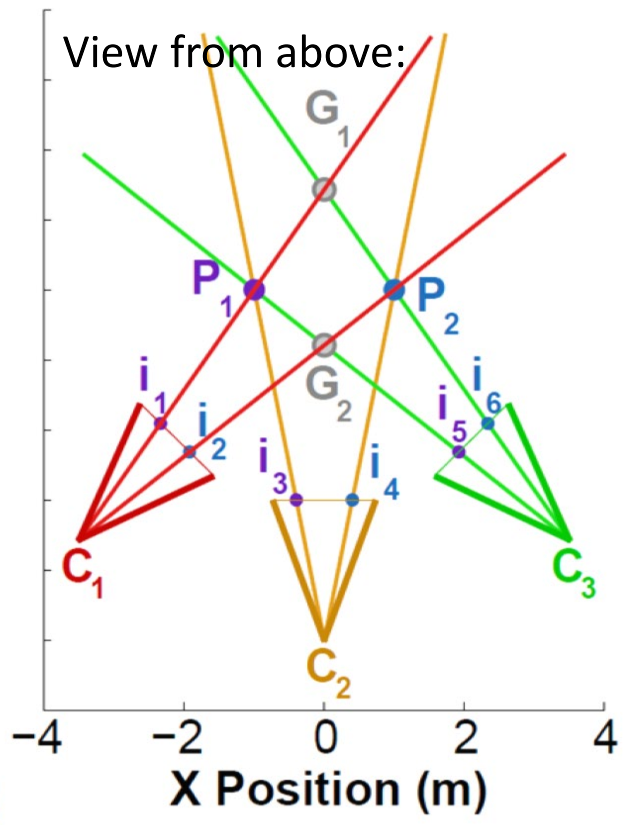


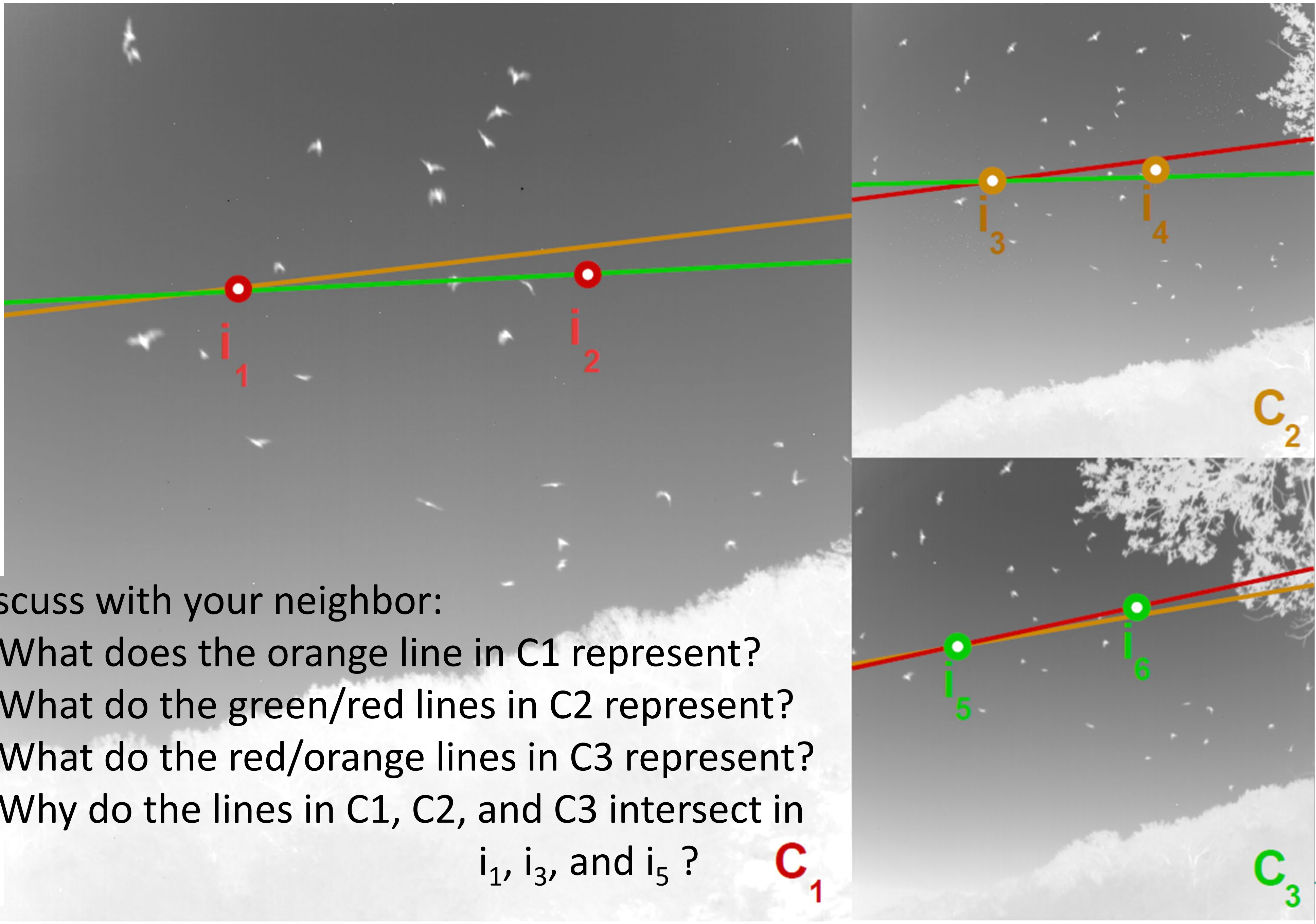
3rd Camera resolves the ambiguity:
G_1 and G_2 are "ghosts" (non-existing points)
P_1 and P_2 are the true scene points





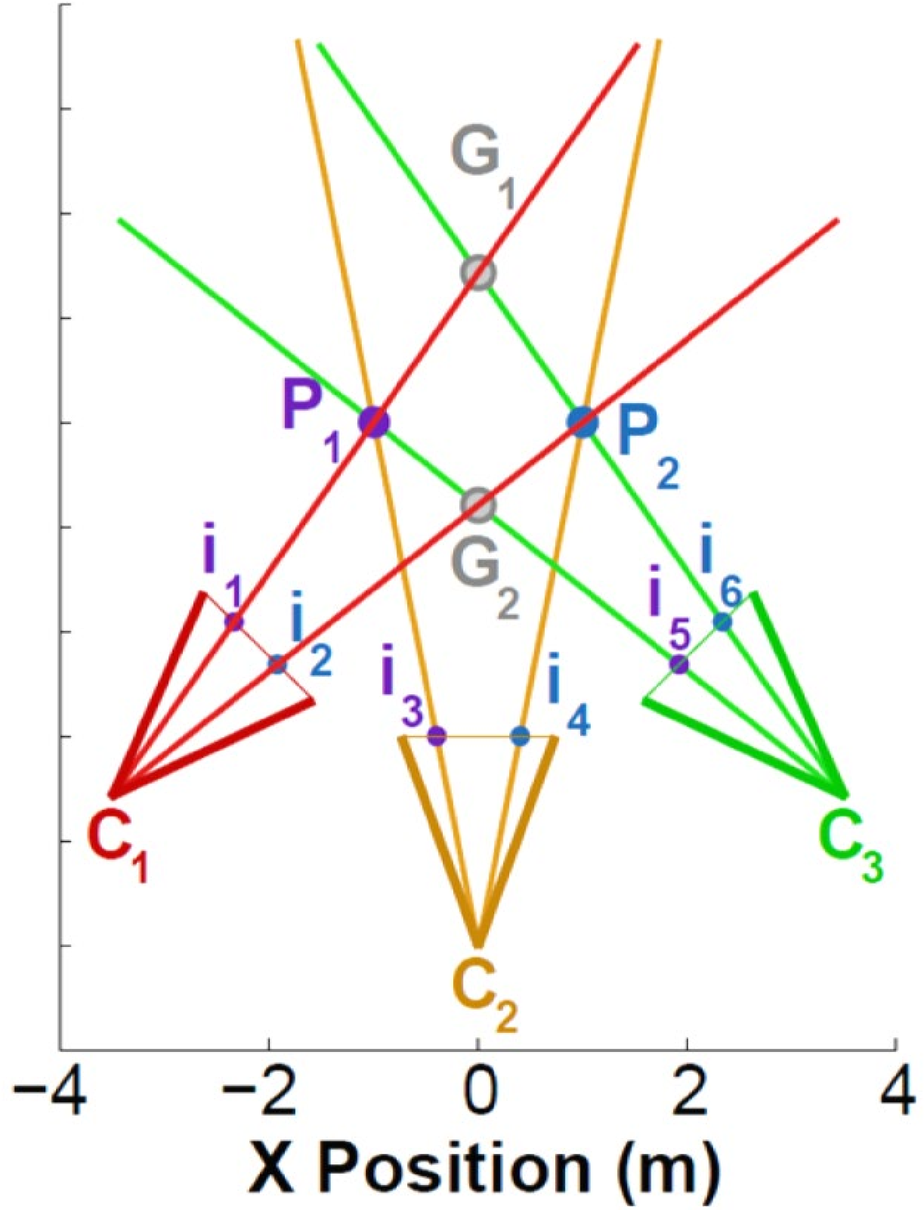
Green line is ray from P_1 into camera C_3.
It appears as an “epipolar line” in the image of camera C_1

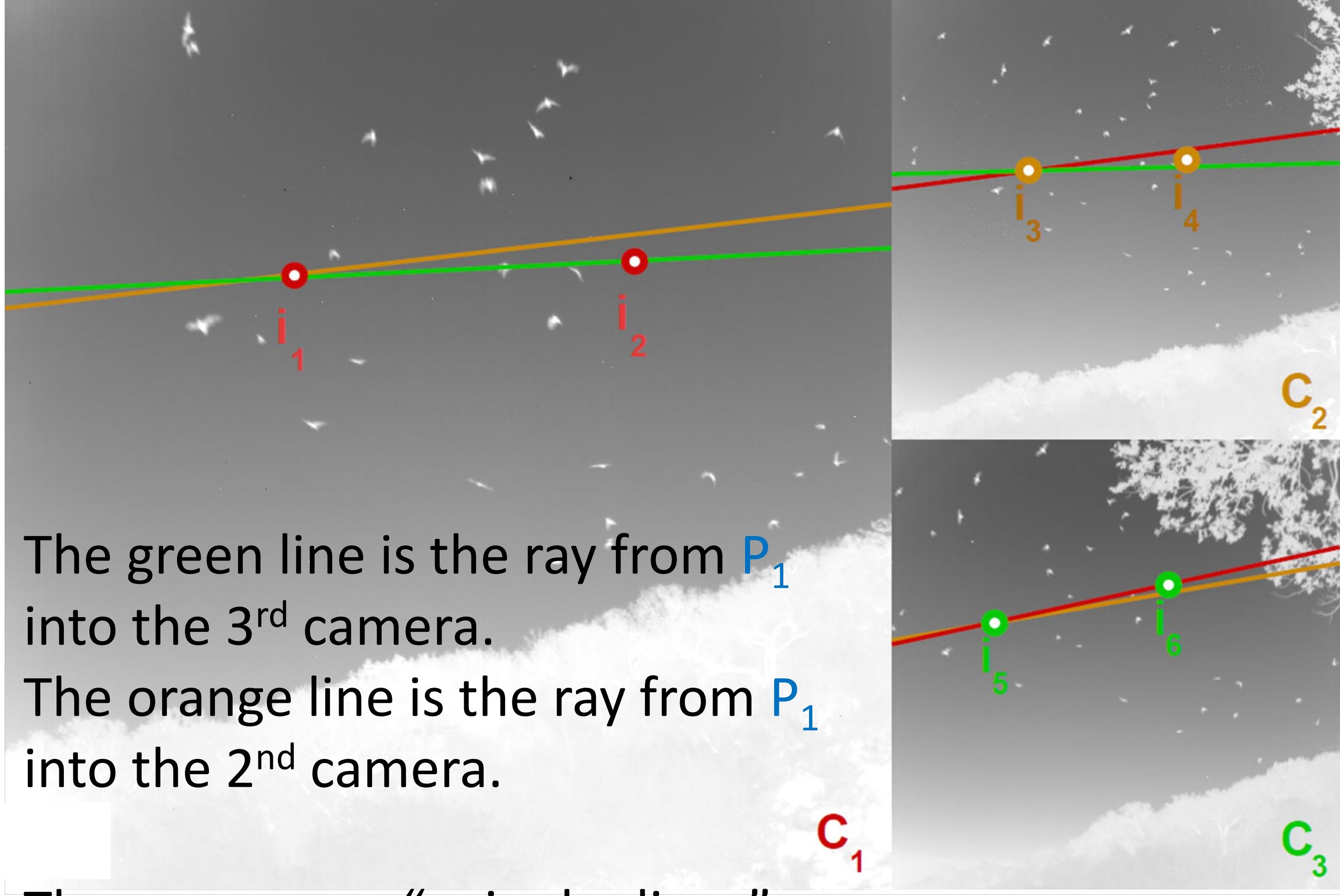




Discuss with your neighbor:

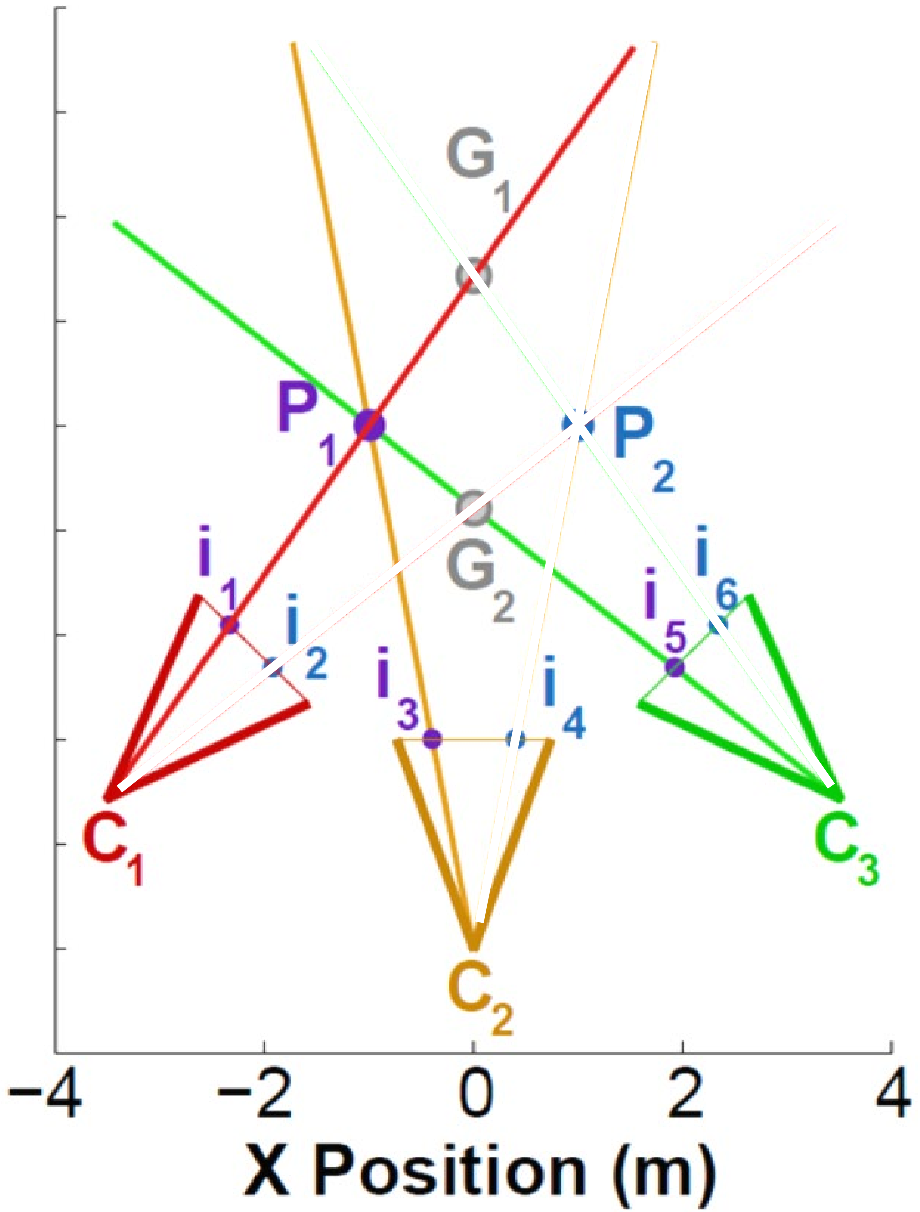
- What does the orange line in C1 represent?
- What do the green/red lines in C2 represent?
- What do the red/orange lines in C3 represent?
- Why do the lines in C1, C2, and C3 intersect in i_1, i_3, and i_5 ?

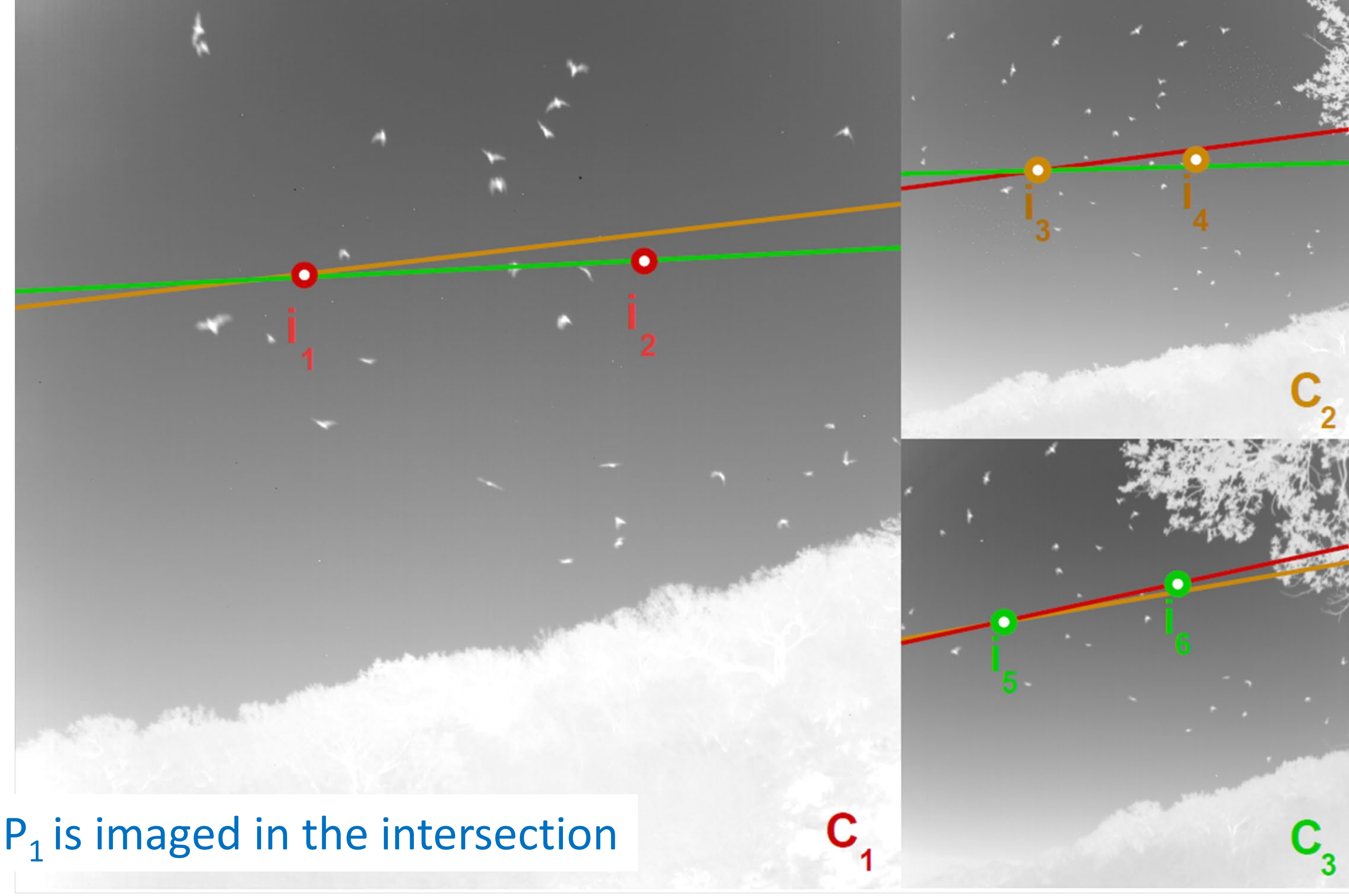




The green line is the ray from P_1 into the 3rd camera.
The orange line is the ray from P_1 into the 2nd camera.

They appear as “epipolar lines” in the image of camera C_1 and must intersect at the same image point i_1.

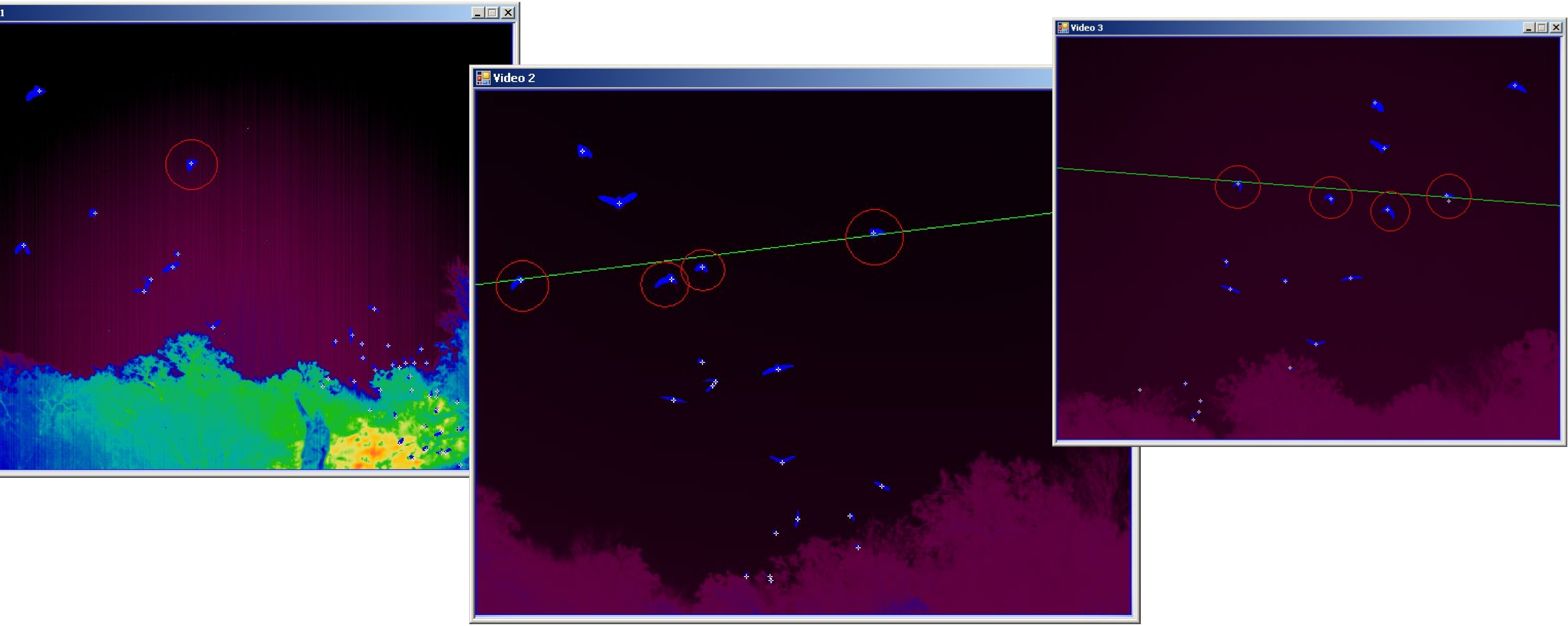




The green and red epipolar lines in the camera C_2 intersect at image point i_3.

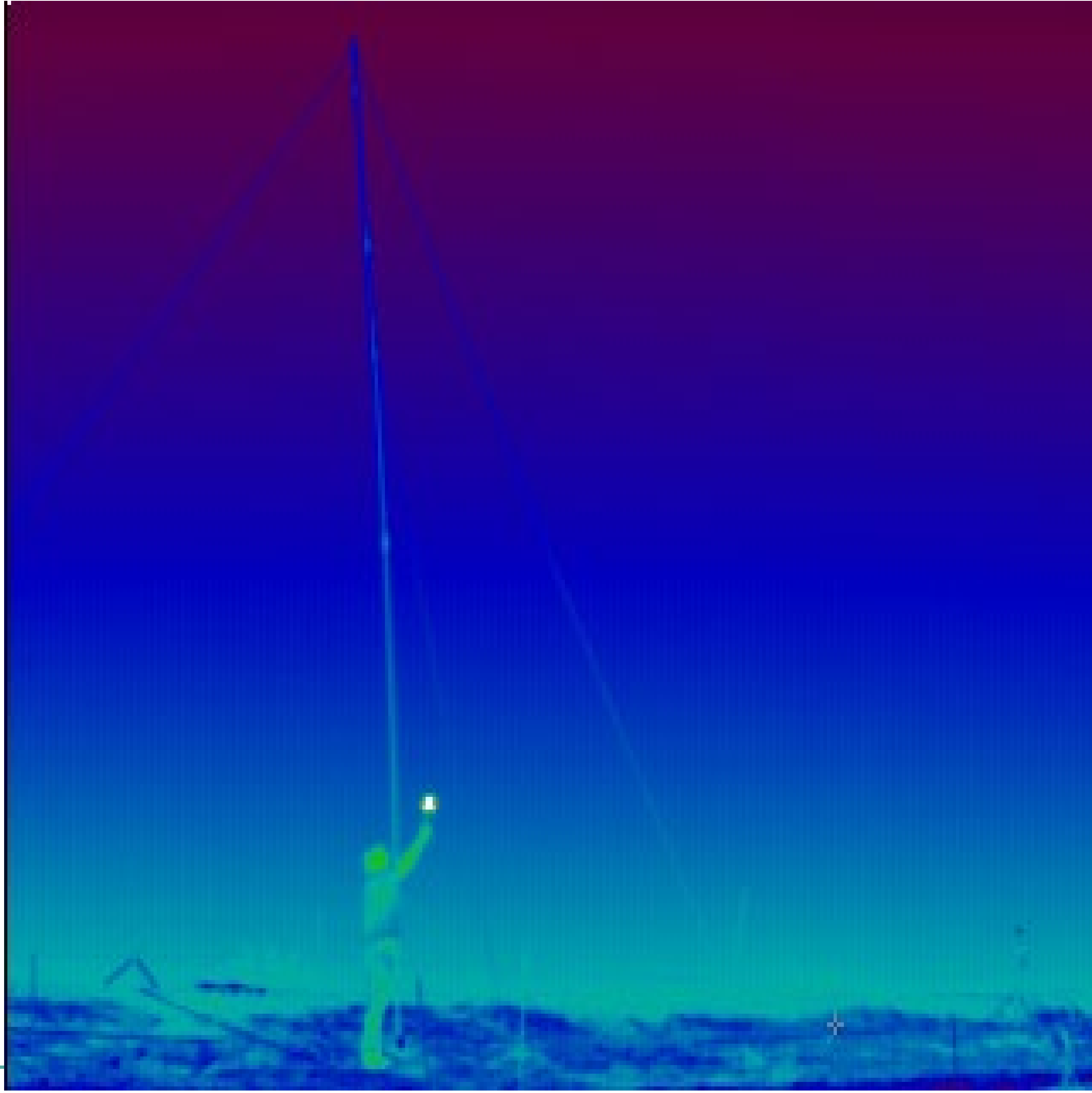
The orange and red epipolar lines in the camera C_3 intersect at image point i_5

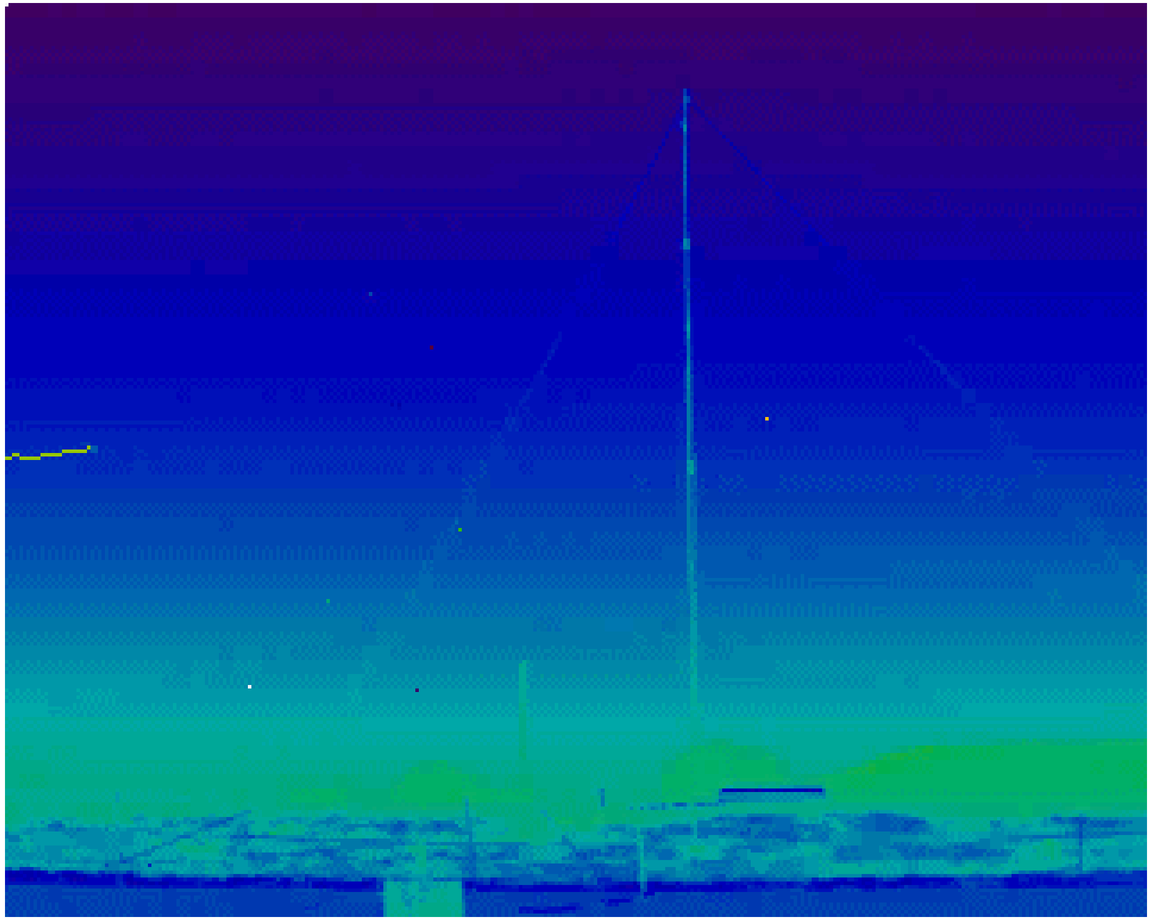
How to use epipolar lines for bat tracking:

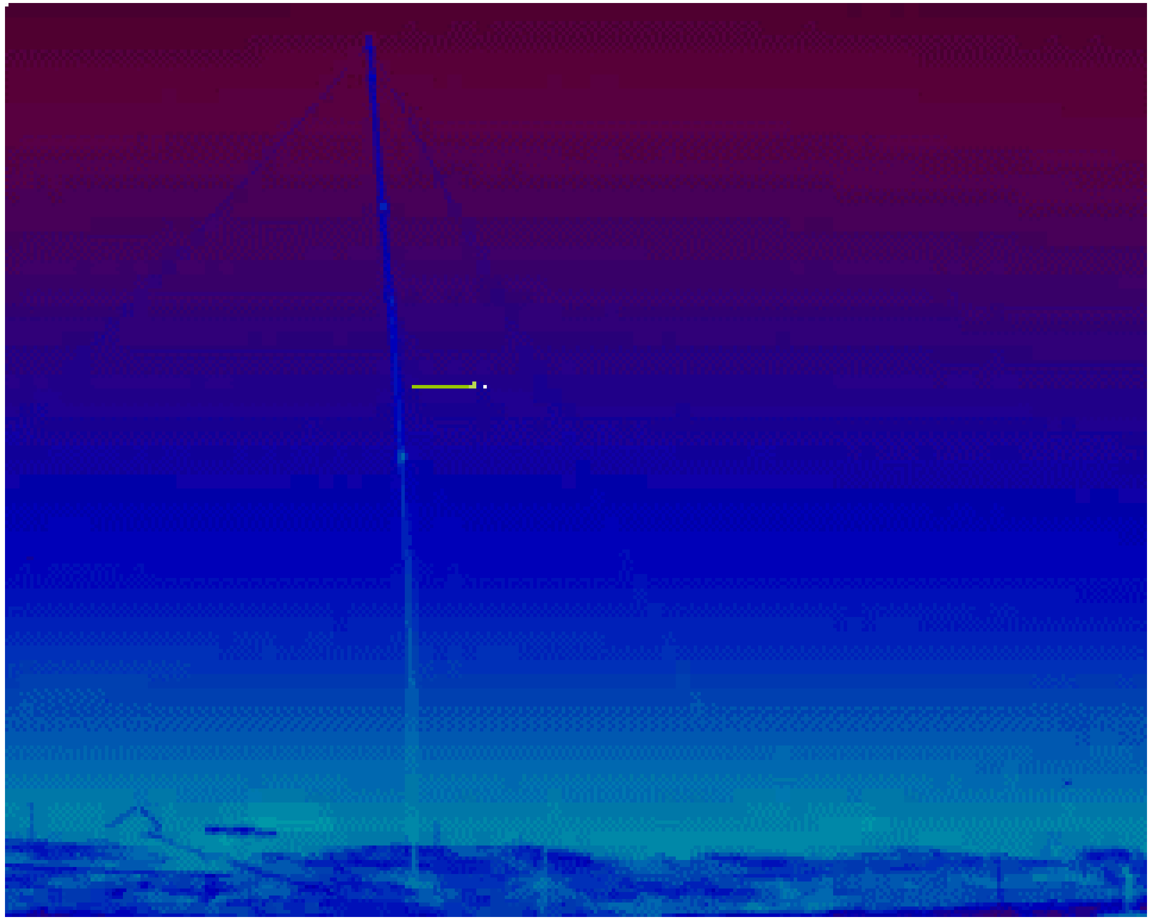


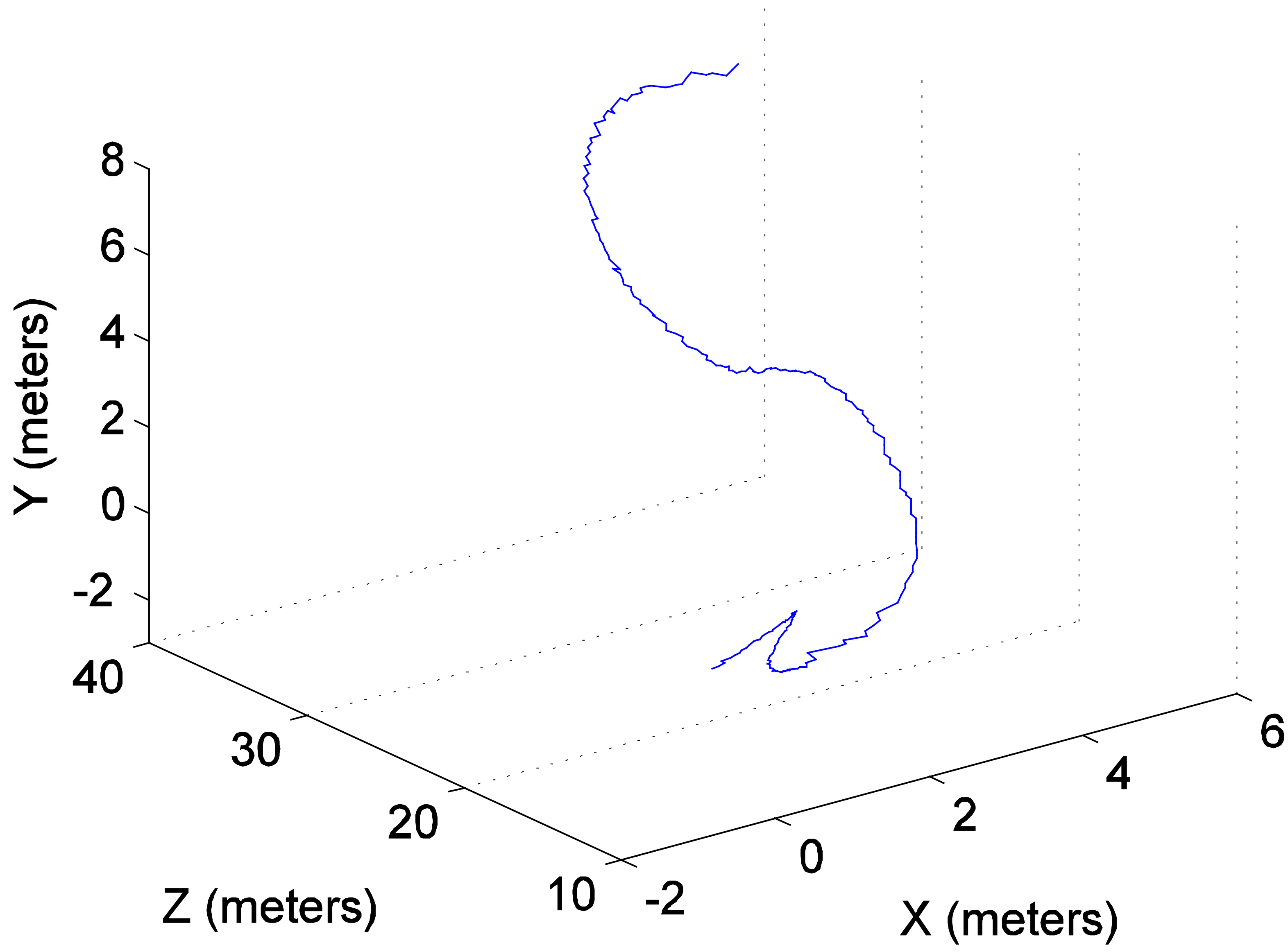
Temporal Calibration

Used a lighter to register the two cameras in time

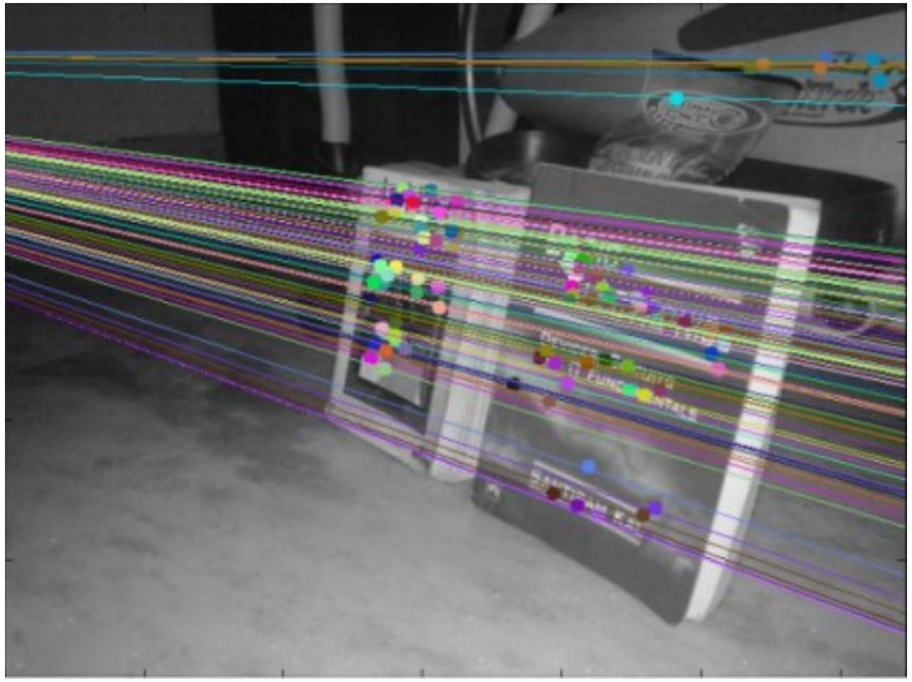




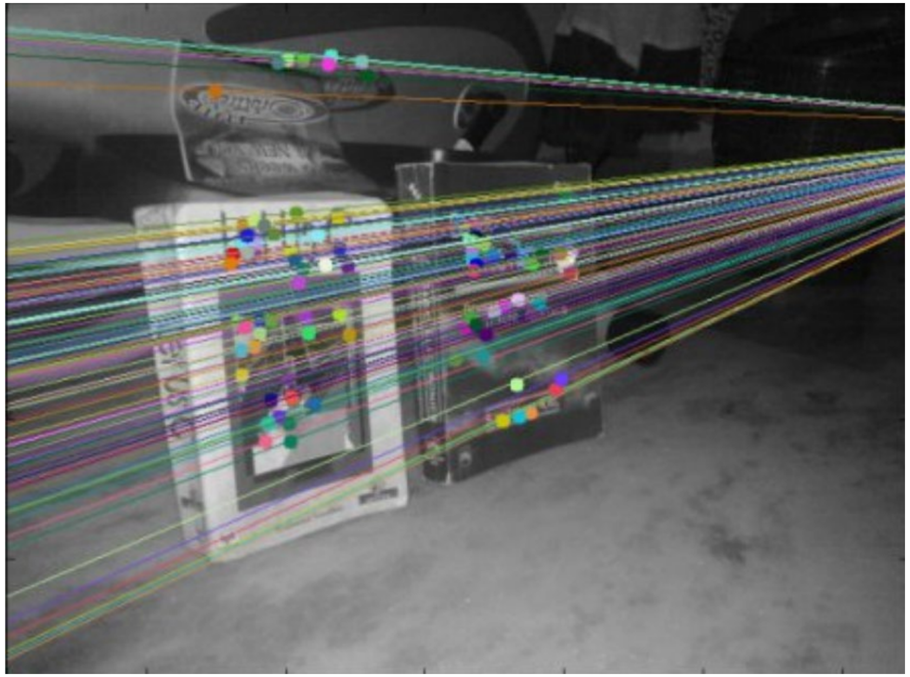



8
6
4
2
0
-2
Y (meters)
40
30
20
10
Z (meters)
-2
0
2
4
6
X (meters)

Epipolar Geometry



left image



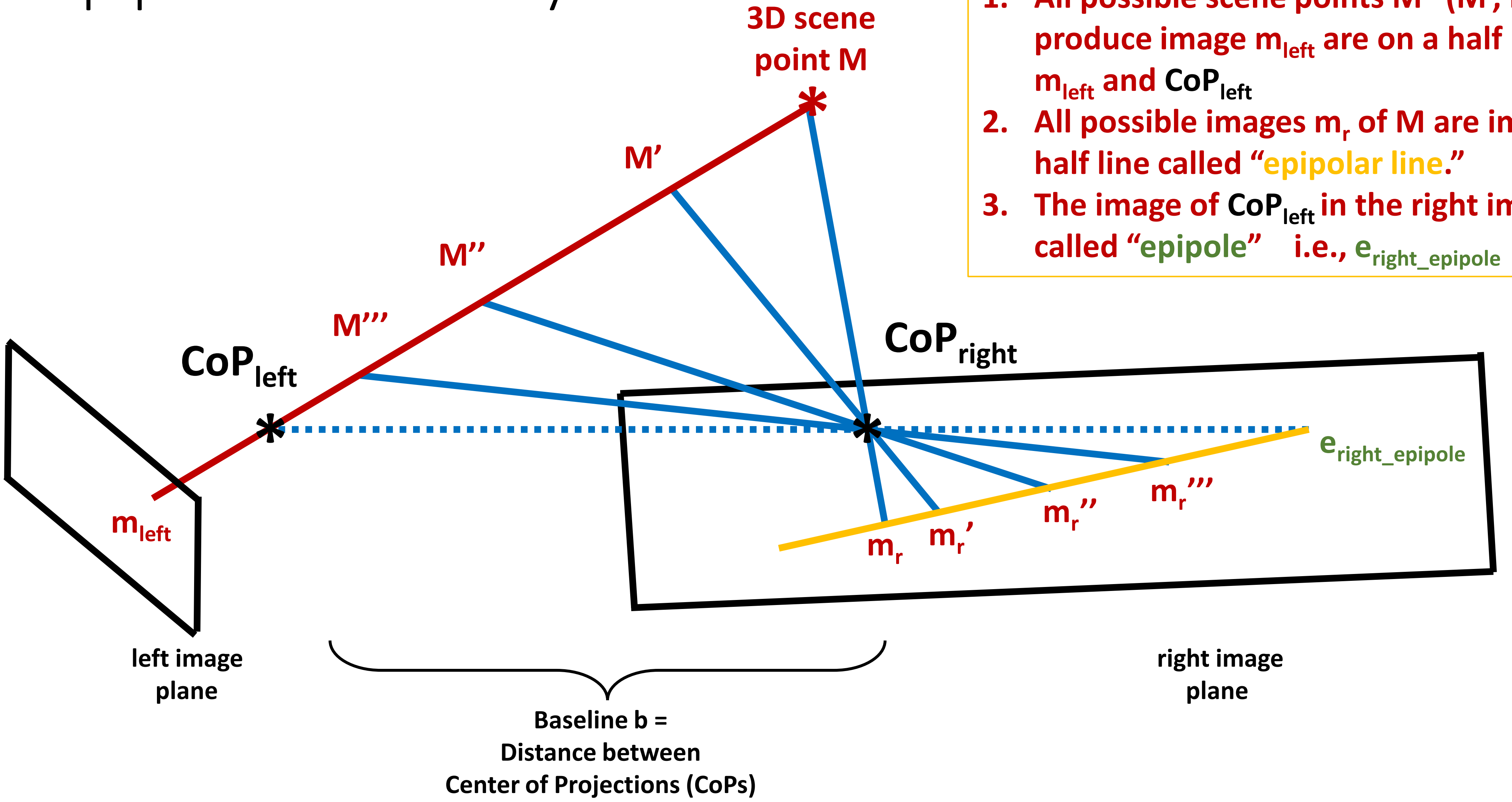
right image

Image Credit: OpenCV.org

Epipolar Geometry

Formal Definition:

1. **All possible scene points M (M', M'', …) that produce image m_{left} are on a half line through m_{left} and CoP_{left}**
2. **All possible images m_r of M are images of this half line called "epipolar line."**
3. **The image of CoP_{left} in the right image plane is called "epipole" i.e., $e_{right\_epipole}$**



Using Epipolar Geometry to Estimate Camera Motion

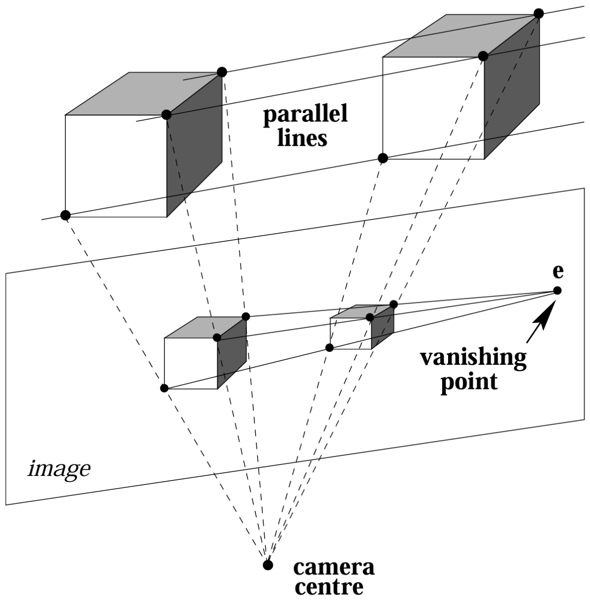


Fig. 9.7. *Under a pure translational camera motion, 3D points appear to slide along parallel rails. The images of these parallel lines intersect in a vanishing point corresponding to the translation direction. The epipole* e *is the vanishing point.*

Sideway Camera Motion:

Epipole = Vanishing Point

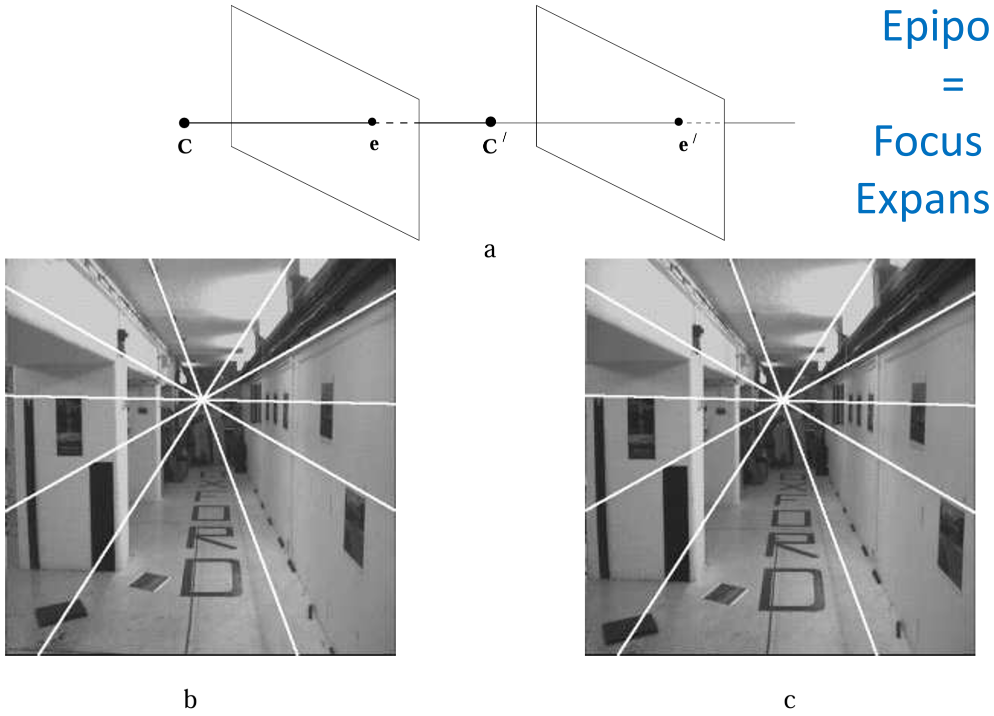


Fig. 9.8. **Pure translational motion.** *(a) under the motion the epipole is a fixed point, i.e. has the same coordinates in both images, and points appear to move along lines radiating from the epipole. The epipole in this case is termed the* Focus of Expansion *(FOE). (b) and (c) the same epipolar lines are overlaid in both cases. Note the motion of the posters on the wall which slide along the epipolar line.*

Forward Camera Motion:

Epipole = Focus of Expansion

Image Credit: Hartley & Zisserman, 2004

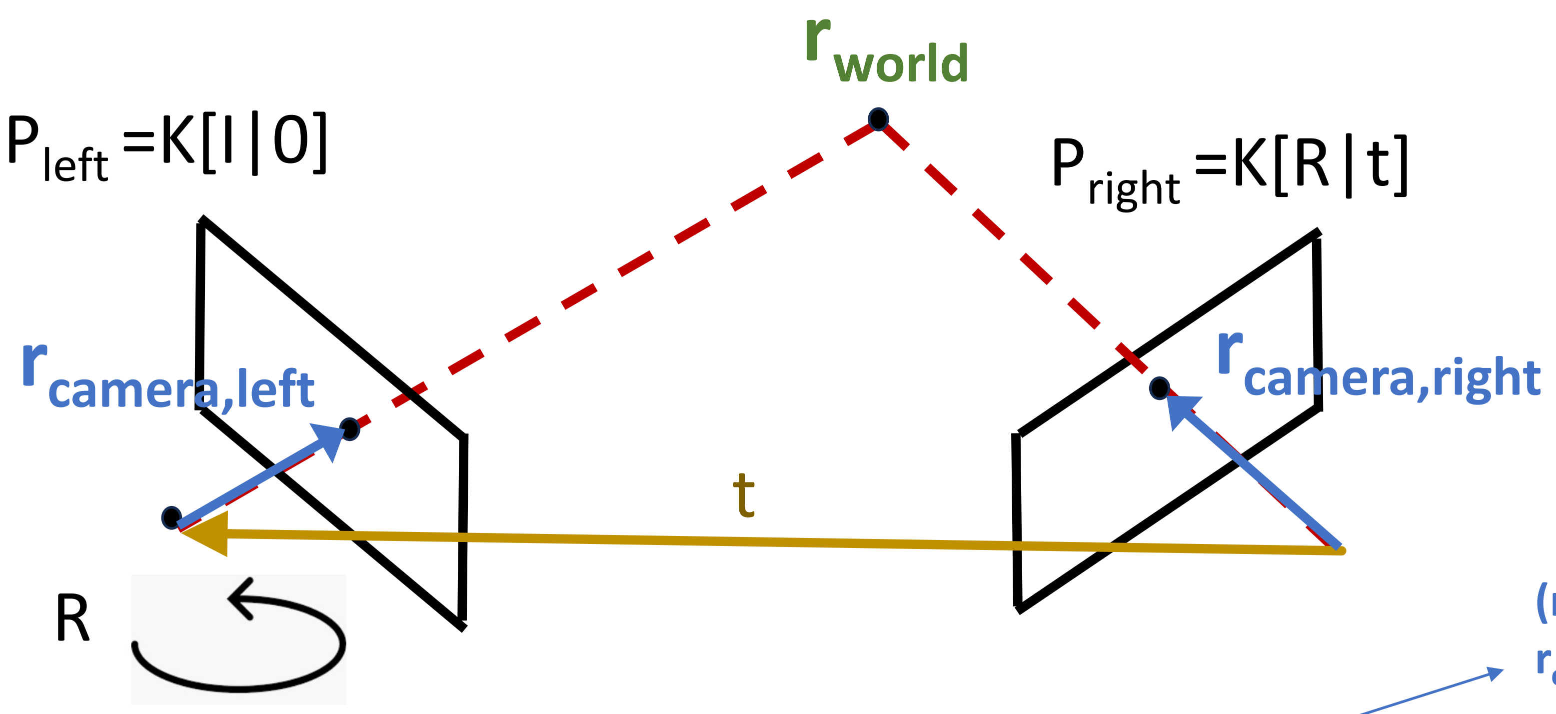
Remember from Linear Algebra:

- 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: $t \times b = [t]_x b$

$$[\mathbf{t}]_x = \begin{pmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{pmatrix}$$

Derivation of the "Fundamental Matrix:"

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



$r_{camera,right} = R\,(r_{camera,left} - t)$ or

$(r_{camera,left} - t)^T = r_{camera,right}^{\ T}\, R$

These three vectors are in the same plane:

$t,\ r_{camera,left},\ r_{camera,right}$

$$\underbrace{r_{camera,left}^{\ T}\,\underbrace{(t \times r_{camera,left})}_{\text{Cross product}}}_{\text{Dot product}} = 0$$

$(r_{camera,left} - t)^T (t \times r_{camera,left}) = 0$

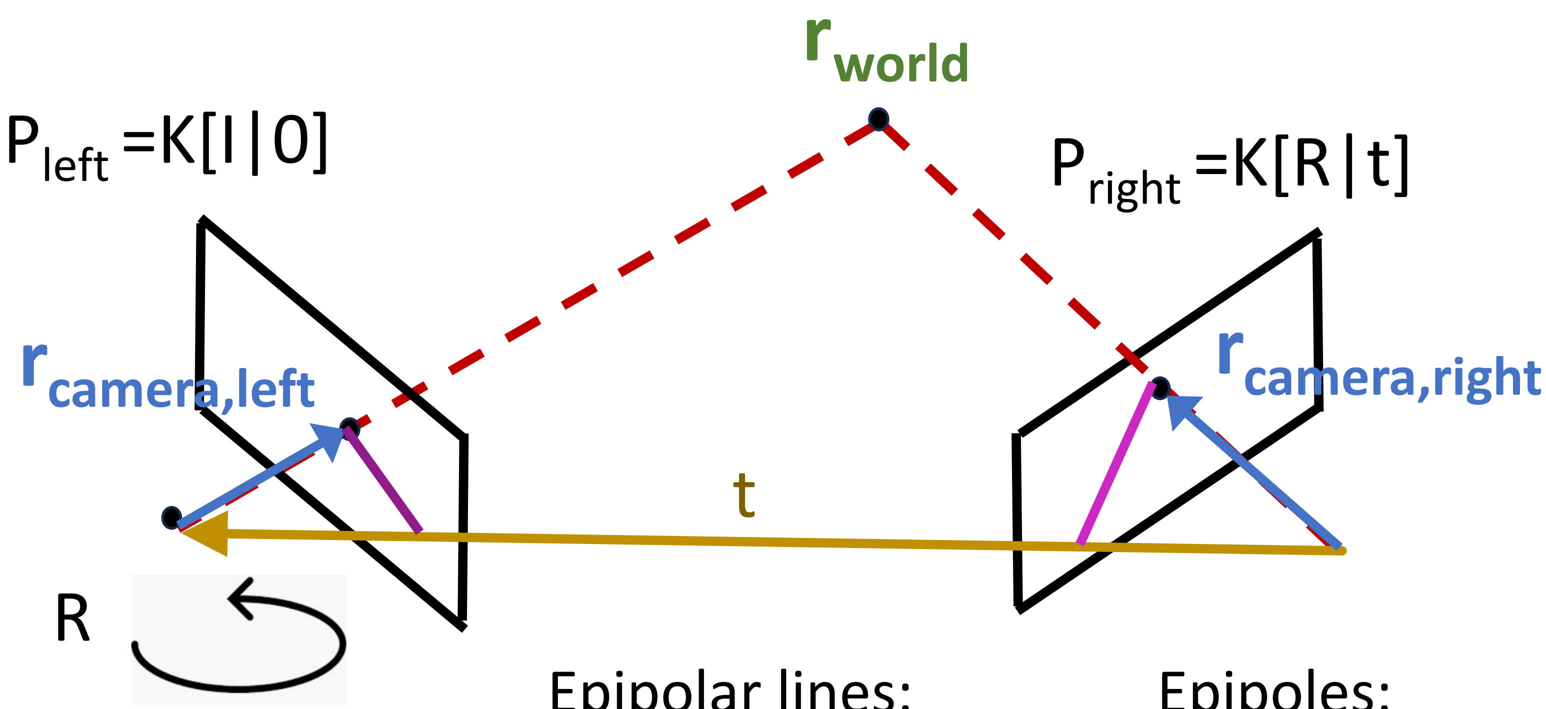
$r_{camera,right}^{\ T}\, R\,(t \times r_{camera,left}) = 0$

$r_{camera,right}^{\ T}\, R\,([t]_x\, r_{camera,left}) = 0$

$r_{camera,right}^{\ T}\,(R\,[t]_x)\, r_{camera,left} = 0$

Here use 4D homogeneous coordinates

Derivation of the "Fundamental Matrix:"



$$\tilde{r}_{camera,right}^{\;T} \underbrace{(R\ [t]_x)}_{\text{“Essential matrix” E}} \tilde{r}_{camera,left} = 0$$

$$\tilde{r}_{camera,right}^{\;T}\ E\ \tilde{r}_{camera,left} = 0$$

$$\tilde{r}_{image,right}^{\;T} \underbrace{K_{right}^{-T}\ E\ K_{left}^{-1}}_{\text{“Fundamental matrix” F}} \tilde{r}_{image,left} = 0$$

$$\boxed{\tilde{r}_{image,right}^{\;T}\ F\ \tilde{r}_{image,left} = 0}$$

Epipolar lines:

$\tilde{L}_{right} = E\ \tilde{r}_{camera,left}$

$\tilde{L}_{left} = E^T\ \tilde{r}_{camera,right}$

Epipoles:

$E\ \tilde{e}_{left} = \mathbf{0}$

$\tilde{e}_{right}^{\;T}\ E = \mathbf{0}$

$\tilde{t}^T\ E = \mathbf{0}$

F is of rank 2

Methods to Solve the Problem of General Binocular Stereo Reconstruction

- 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

Methods to Solve the Problem of General Binocular Stereo Reconstruction

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

$$\tilde{\mathbf{r}}^T_{\text{camera,right}} F \, \tilde{\mathbf{r}}_{\text{camera,left}} = (x_r, y_r, 1) \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \begin{pmatrix} x_l \\ y_l \\ 1 \end{pmatrix} = 0$$

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

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:

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

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

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 Σ.

Hartley’s Normalized 8-point algorithm

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

$$
\begin{array}{cccccccccc}
 & 10^6 & 10^6 & 10^3 & & \cdots & & & & 1 \\
\mathbf{Uf} = & \multicolumn{9}{c}{\begin{pmatrix} & & & & \vdots & & & & \\ x_r x_l & x_r y_l & x_r & y_r x_l & y_r y_l & y_r & x_l & y_l & 1 \\ & & & & \vdots & & & & \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{pmatrix} = 0}
\end{array}
$$

This causes numerical instability.

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

Compute F. Enforce singularity. Scale back entries. Compute R&t.

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 $\mathbf{t}$, $\mathbf{r}_{camera,left}$, $\mathbf{r}_{camera,right}$ 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

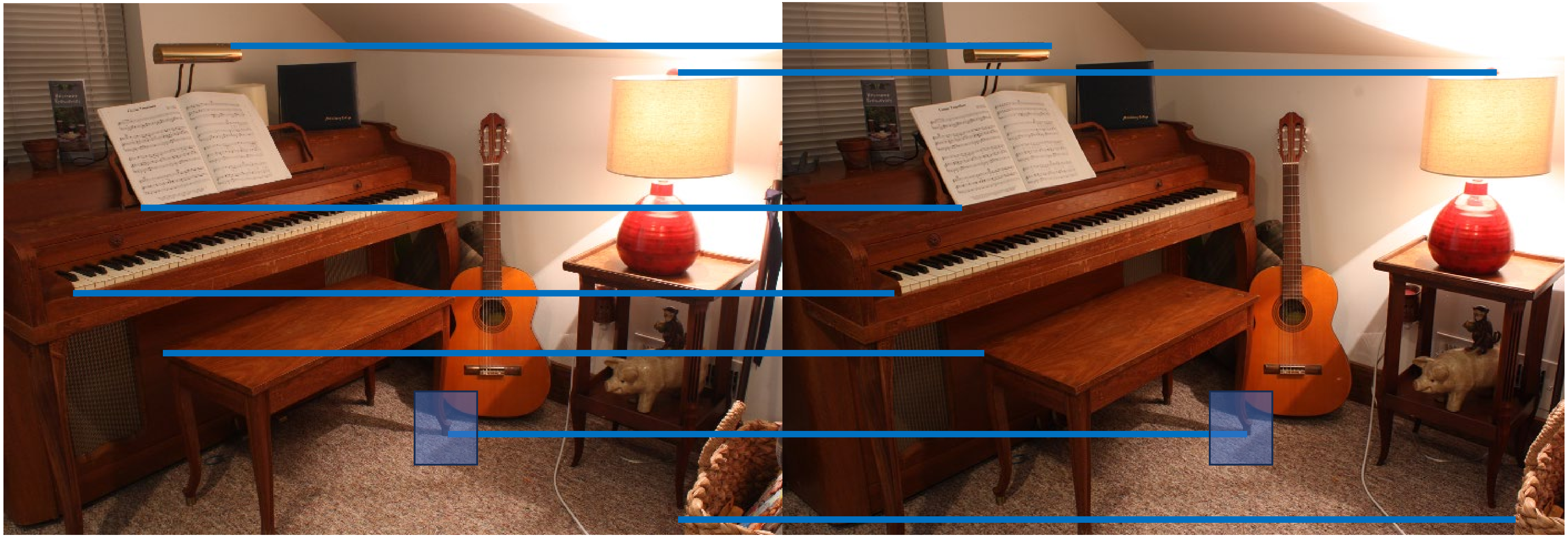
Image Credit: Scharstein, 2014

left image

right image

Image Credit: OpenCV.org

Finding Matching Points: Follow Epipolar Lines & Template Match



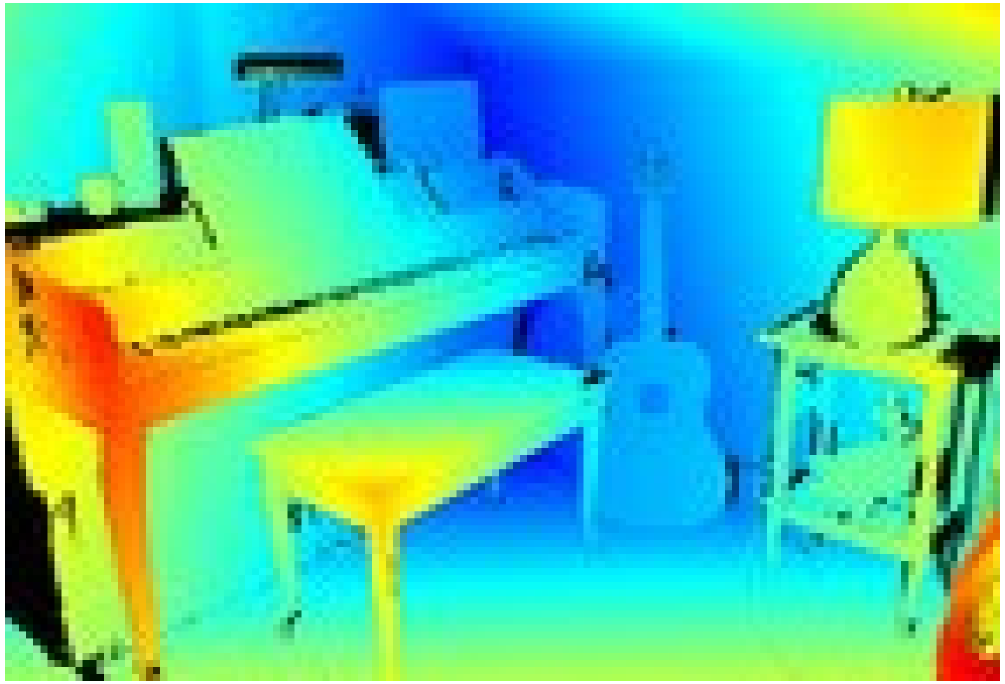
left image

right image

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)

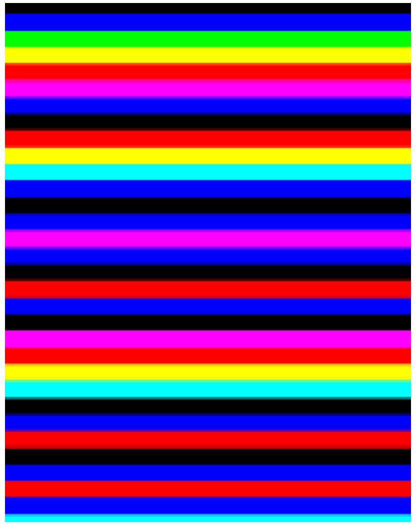
Result of Binocular Stereo Matching: Depth Map

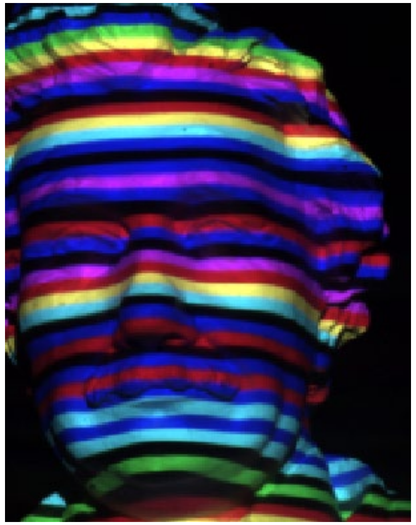


$$Z = bf/\delta$$

http://vision.middlebury.edu/stereo/data/scenes2014/

Parallel Optical Axes & Active Stereo with Structured Light

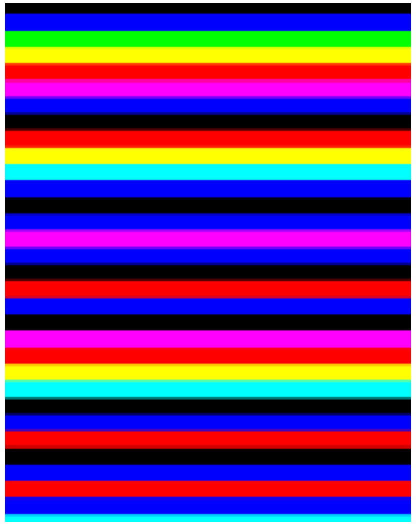




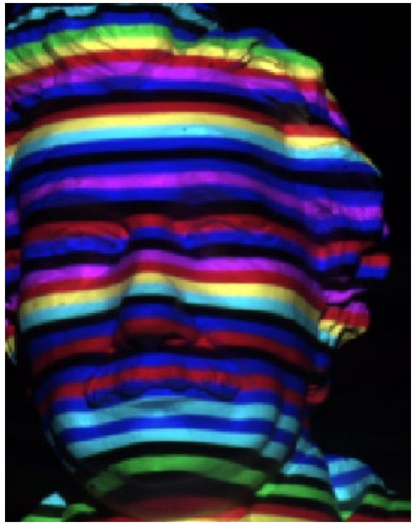
L. Zhang, B. Curless, and S. M. Seitz. [Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming.]() *3DPVT* 2002

Parallel Optical Axes & Active Stereo with Structured Light

Illuminant



Illuminant on Scene



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

Apple Face ID

L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. *3DPVT* 2002

Active stereo with structured light

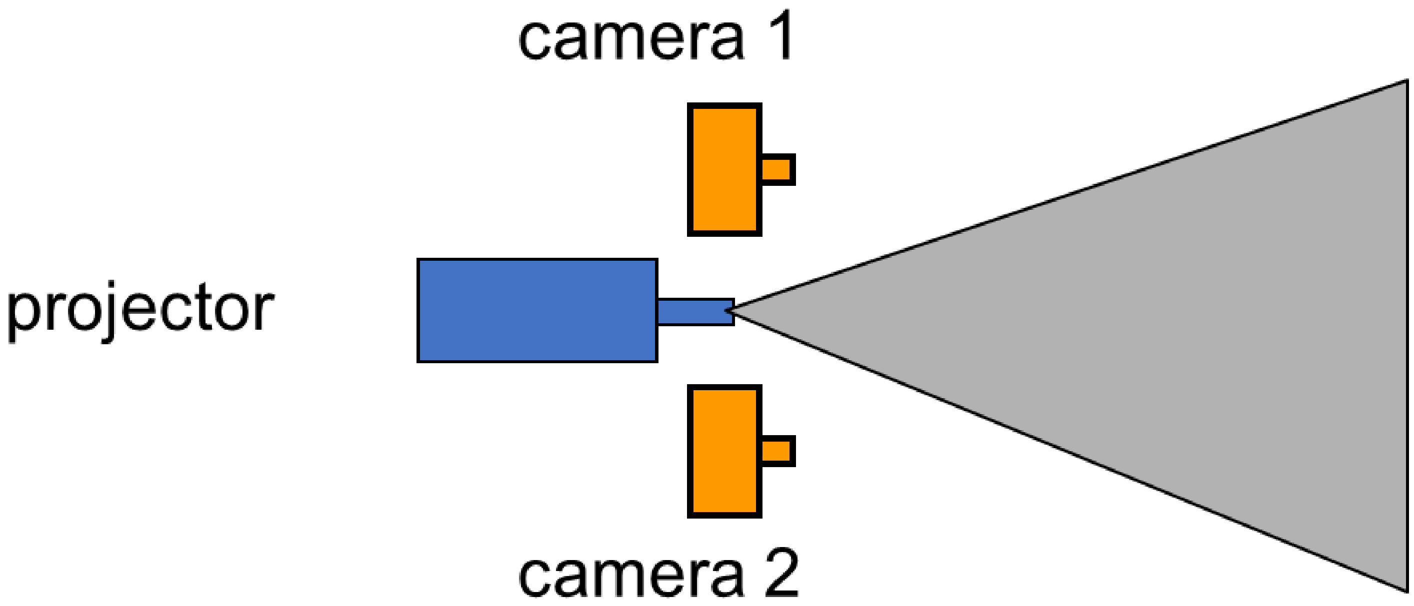
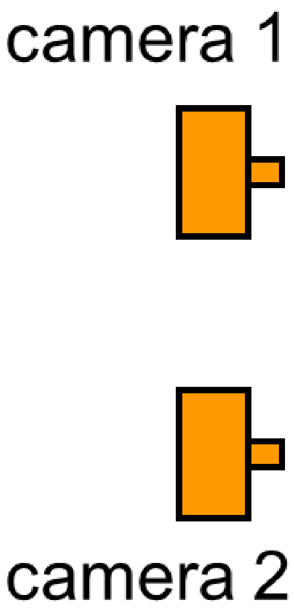


Image credit: Li Zhang

Active stereo with structured light

View without structured light



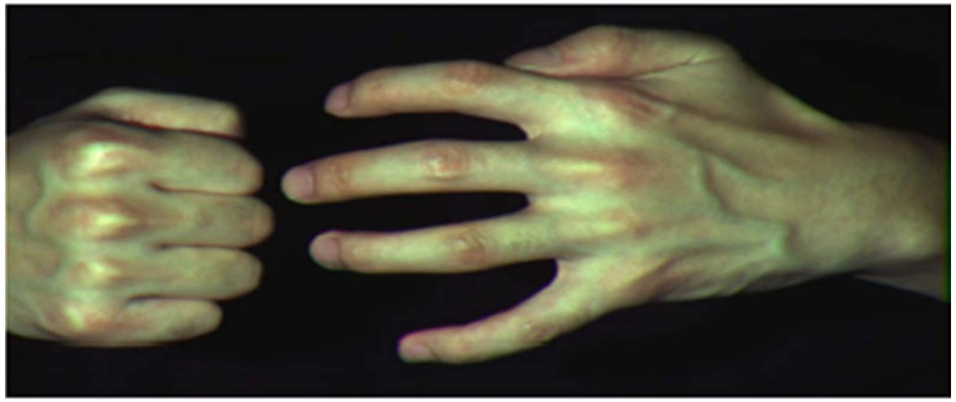
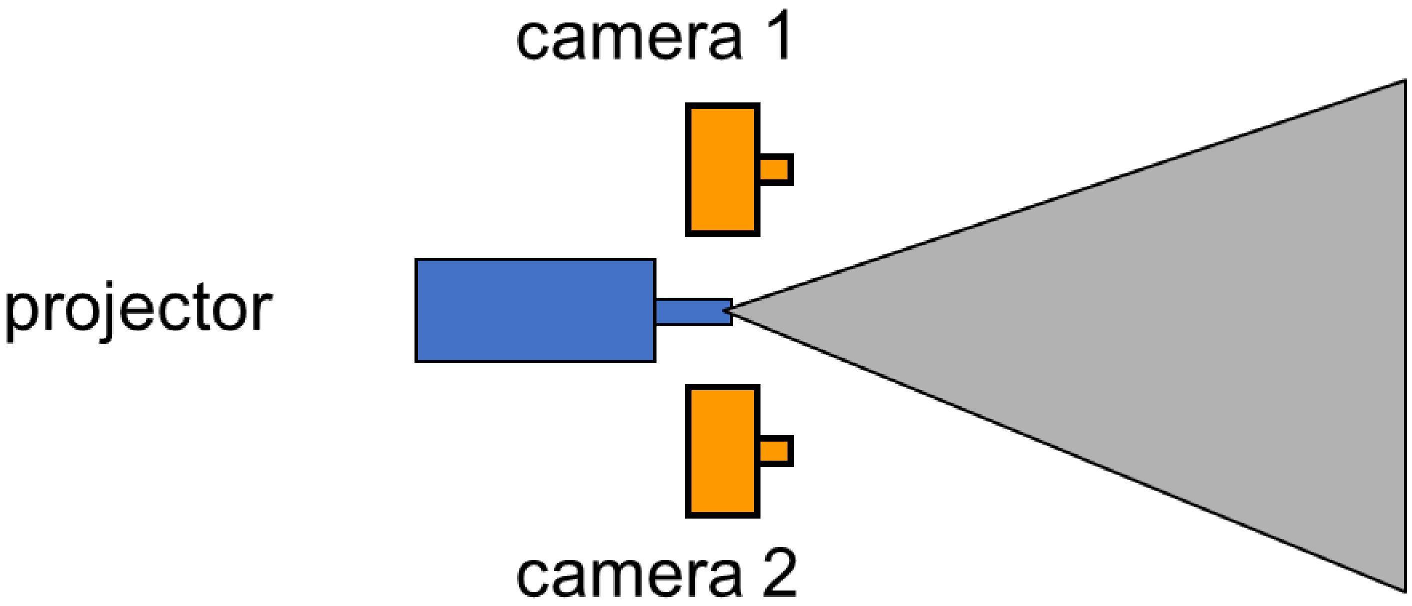
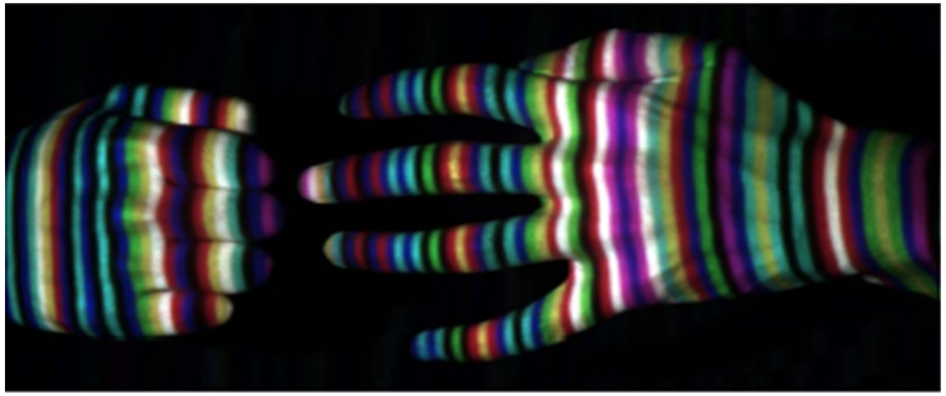


Image credit: Li Zhang

Active stereo with structured light



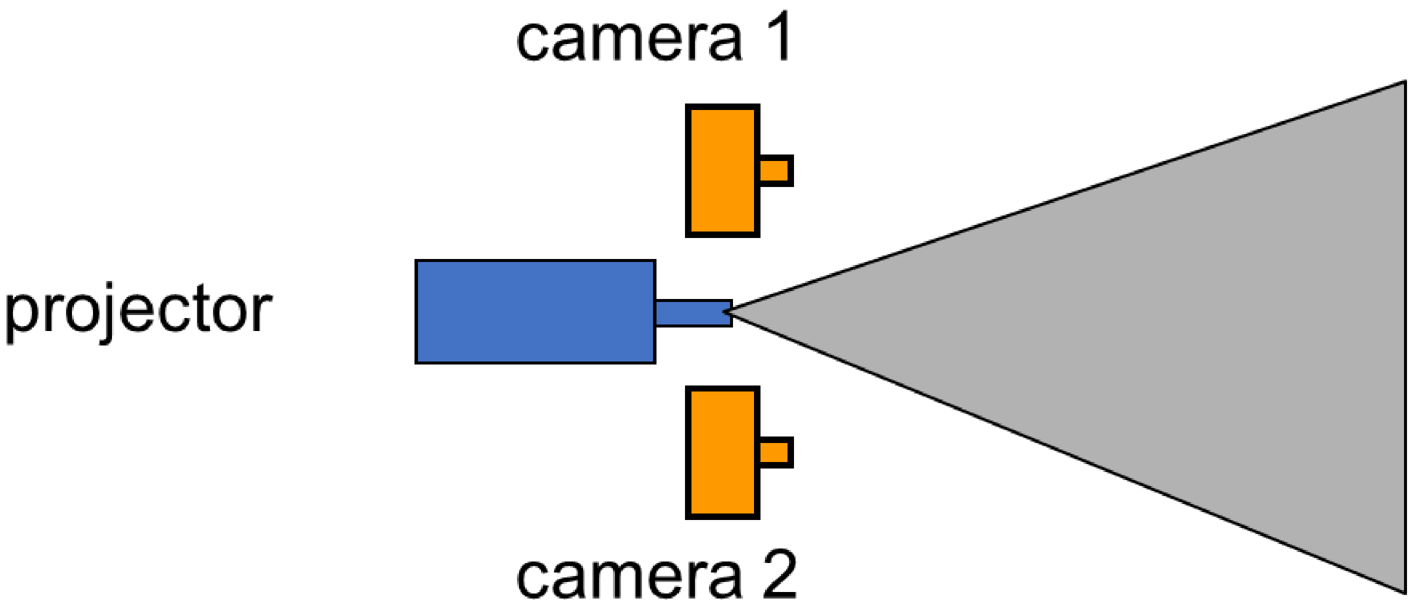


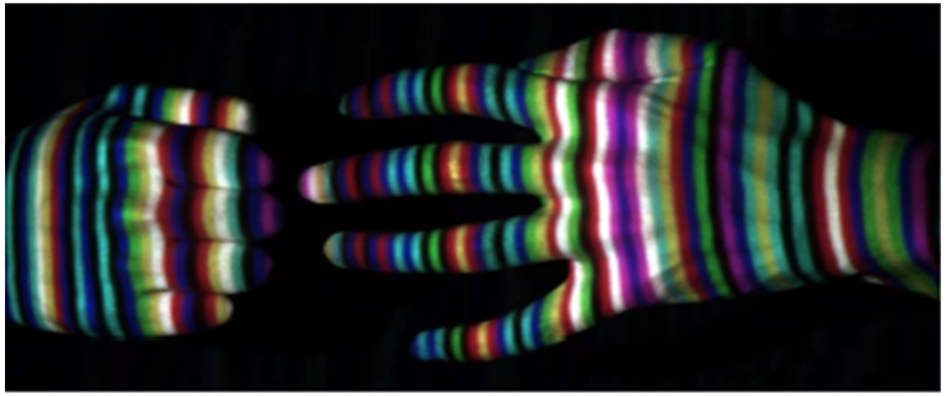
Project “structured” light patterns onto the object

simplifies the correspondence problem

Image credit: Li Zhang

Active stereo with structured light





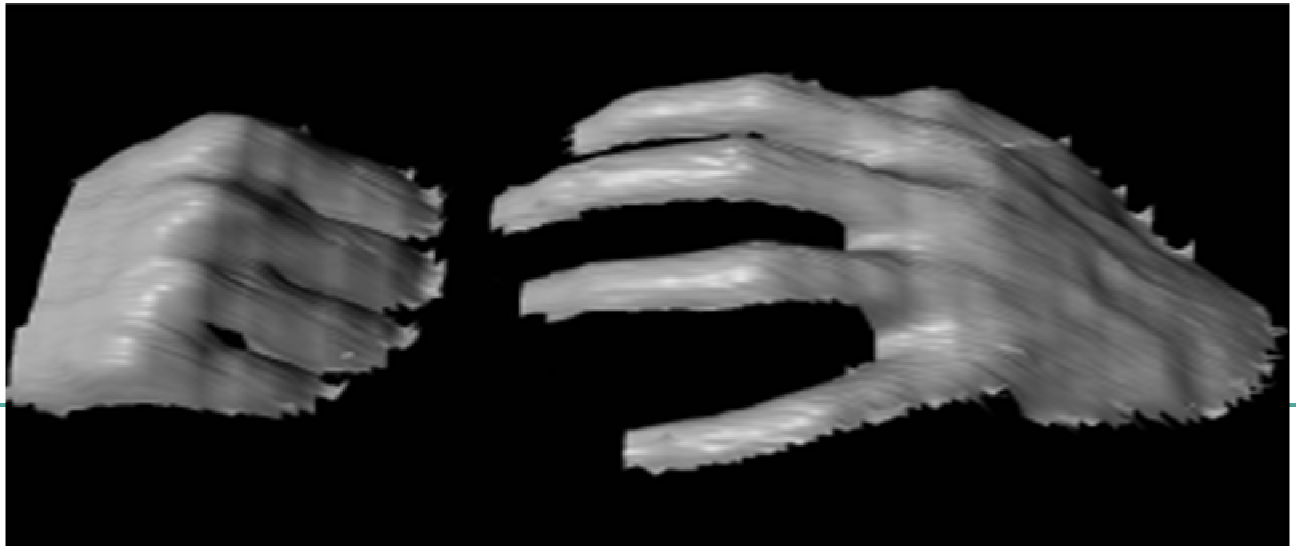
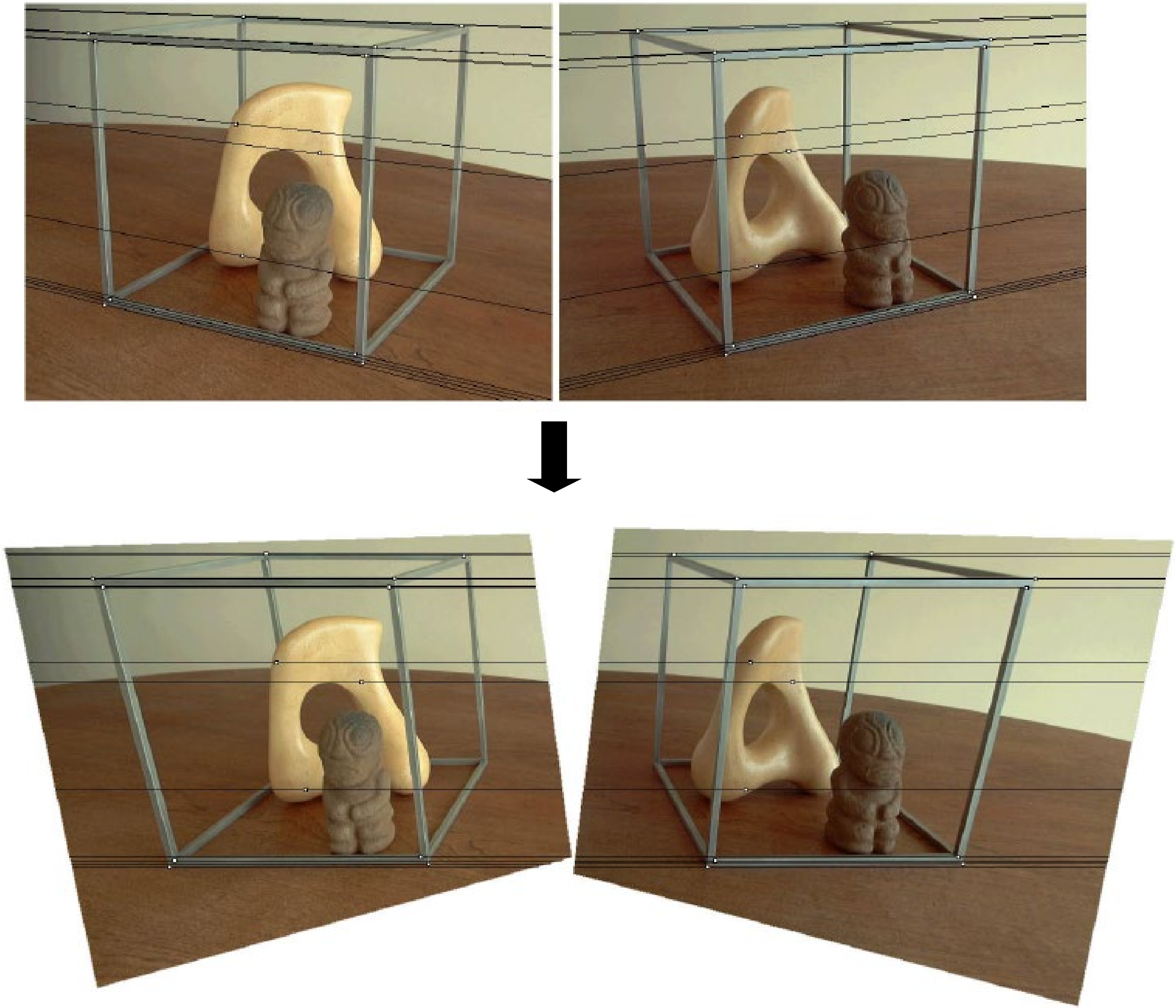


Image credit: Li Zhang

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



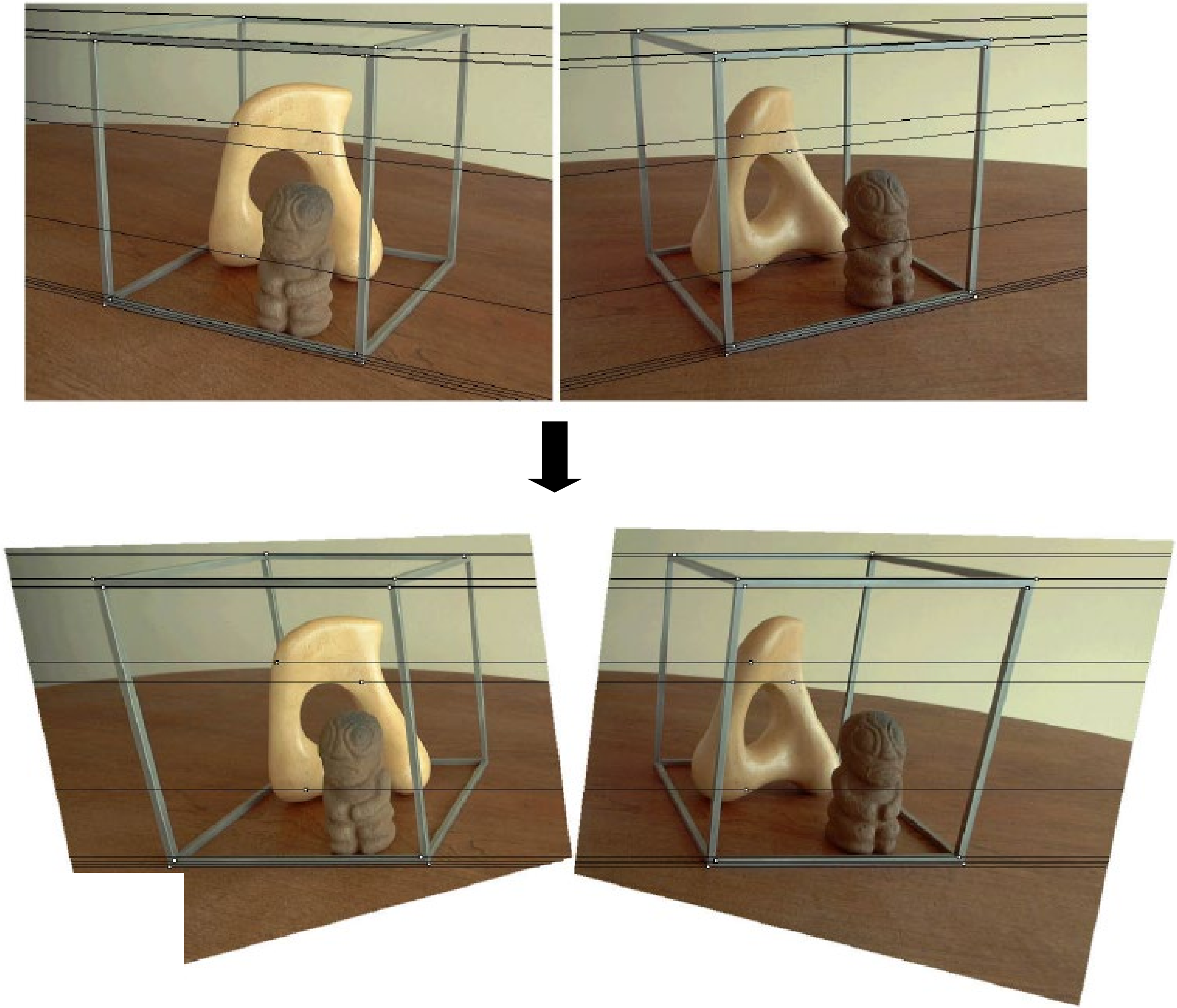
Why?

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

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

Least Squares Method:

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

p = patch
size of patch **p**: tradeoff
too small instability
too large   smearing

Algorithm:
Use current estimate of disparity δ to warp
Then solve LSM & update disparity

Debevec, Taylor, & Malik. [Modeling and Rendering Architecture from Photographs.] SIGGRAPH 1996.

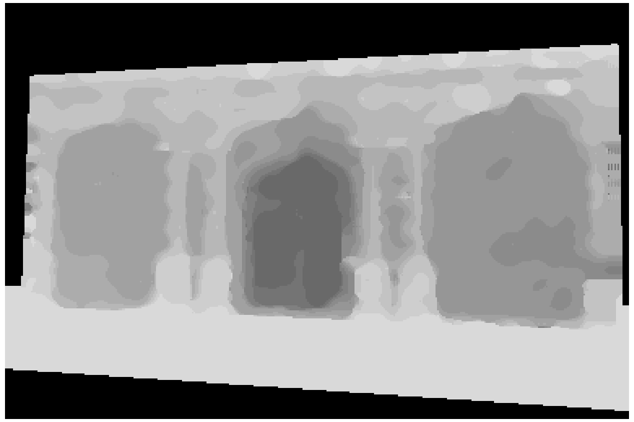
key image

offset image

Debevec, Taylor, & Malik. [Modeling and Rendering Architecture from Photographs.] SIGGRAPH 1996.

key image

warped offset image



depth map

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 δ
   - 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 δ
   - 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)

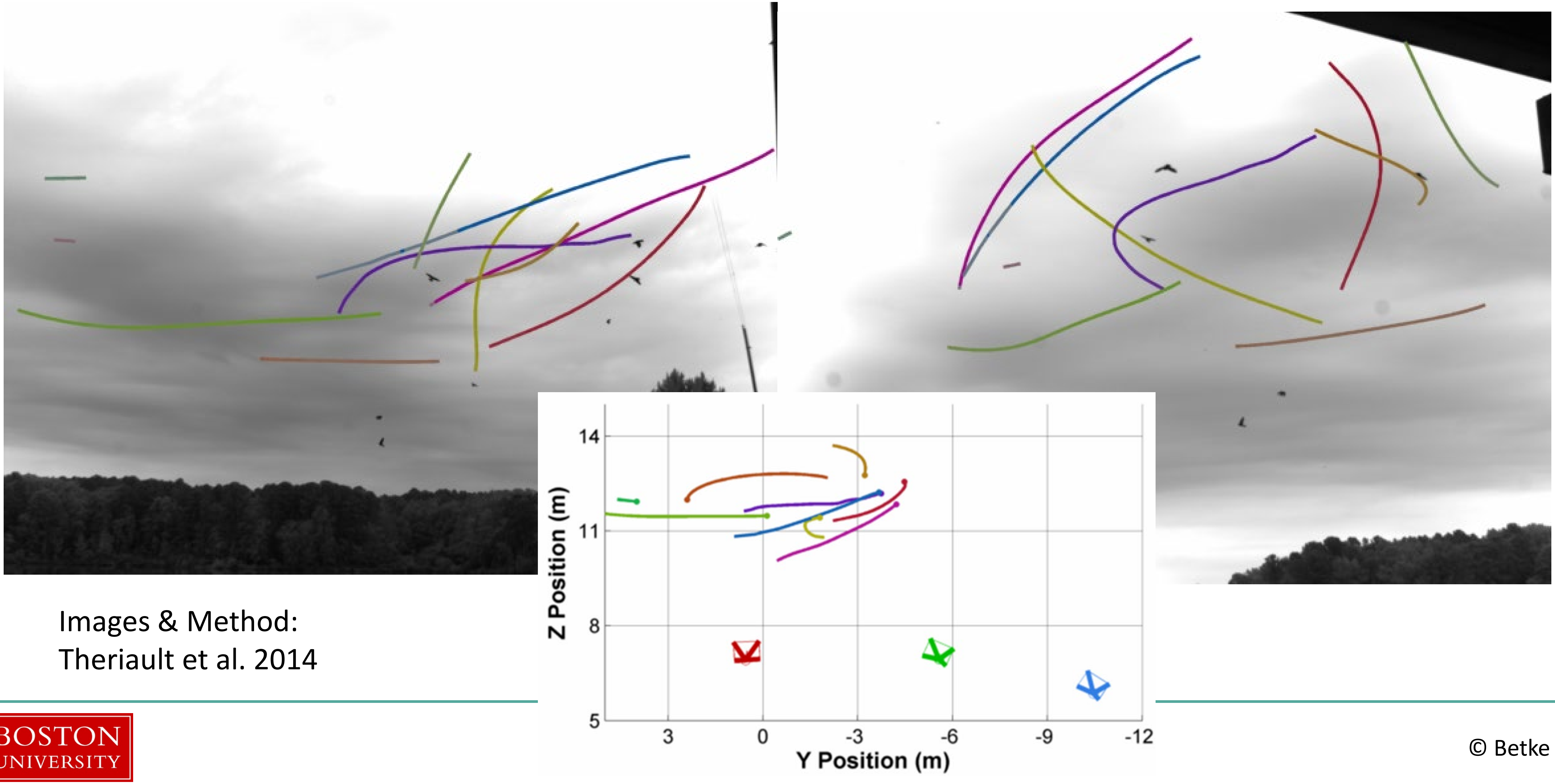
Identify wand position in all views
Take advantage of knowing the dimensions of the wand

Estimate R and r_0

Images & Method:
Theriault et al. 2014

Binocular Stereo Solution Path: “Strong Calibration”

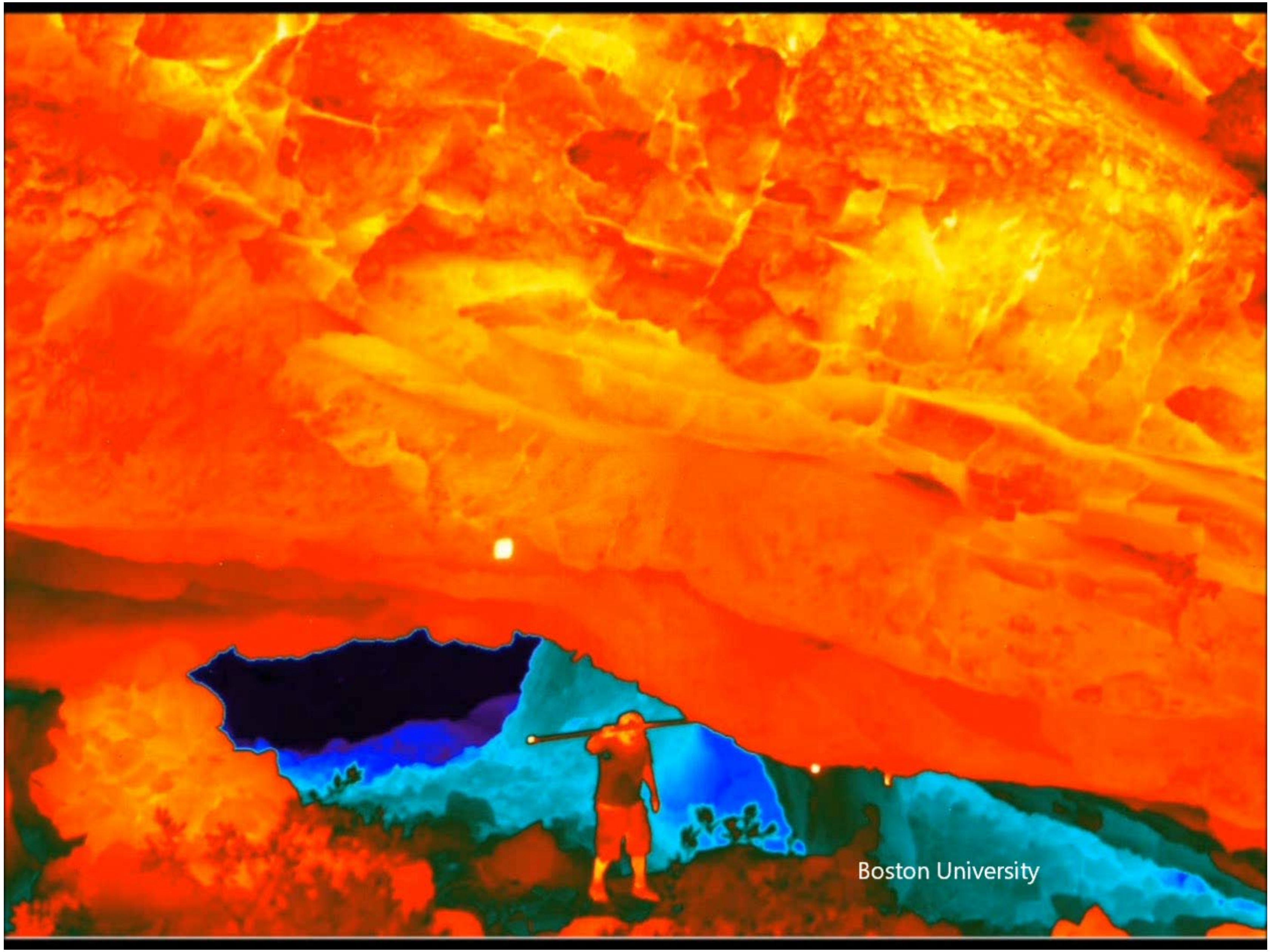
Binocular Stereo for 3D Bird Flight Analysis



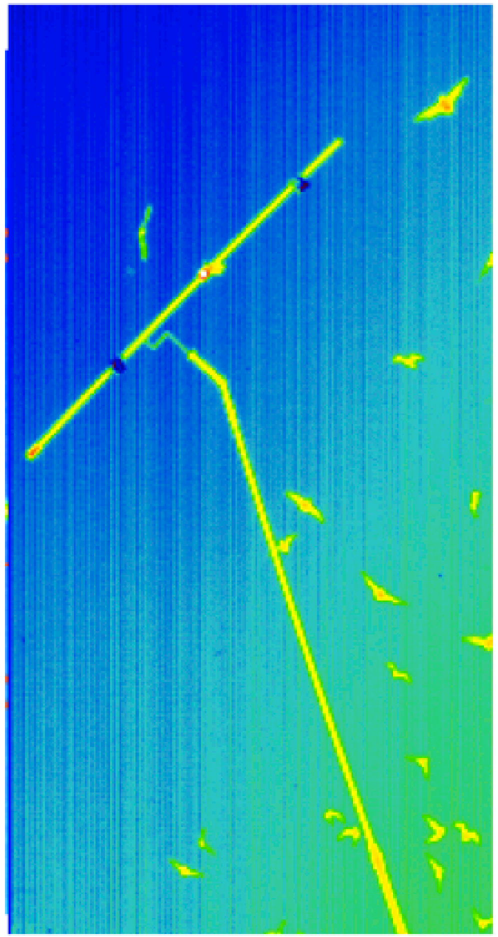
Images & Method:
Theriault et al. 2014

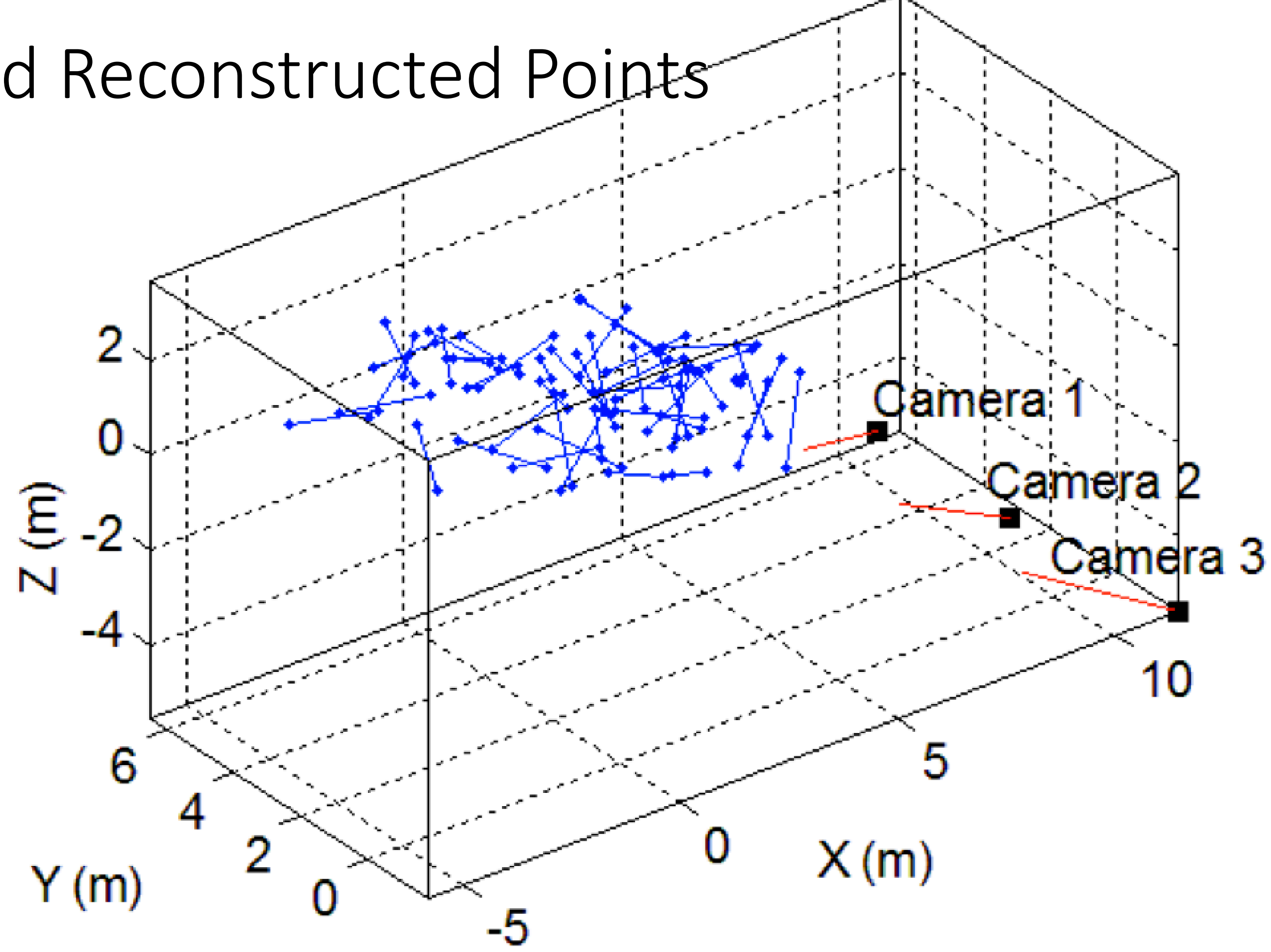
Calibration tool for thermal infrared cameras & Large Observation Spaces

Calibration tool with heat and ice packs


Boston University

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

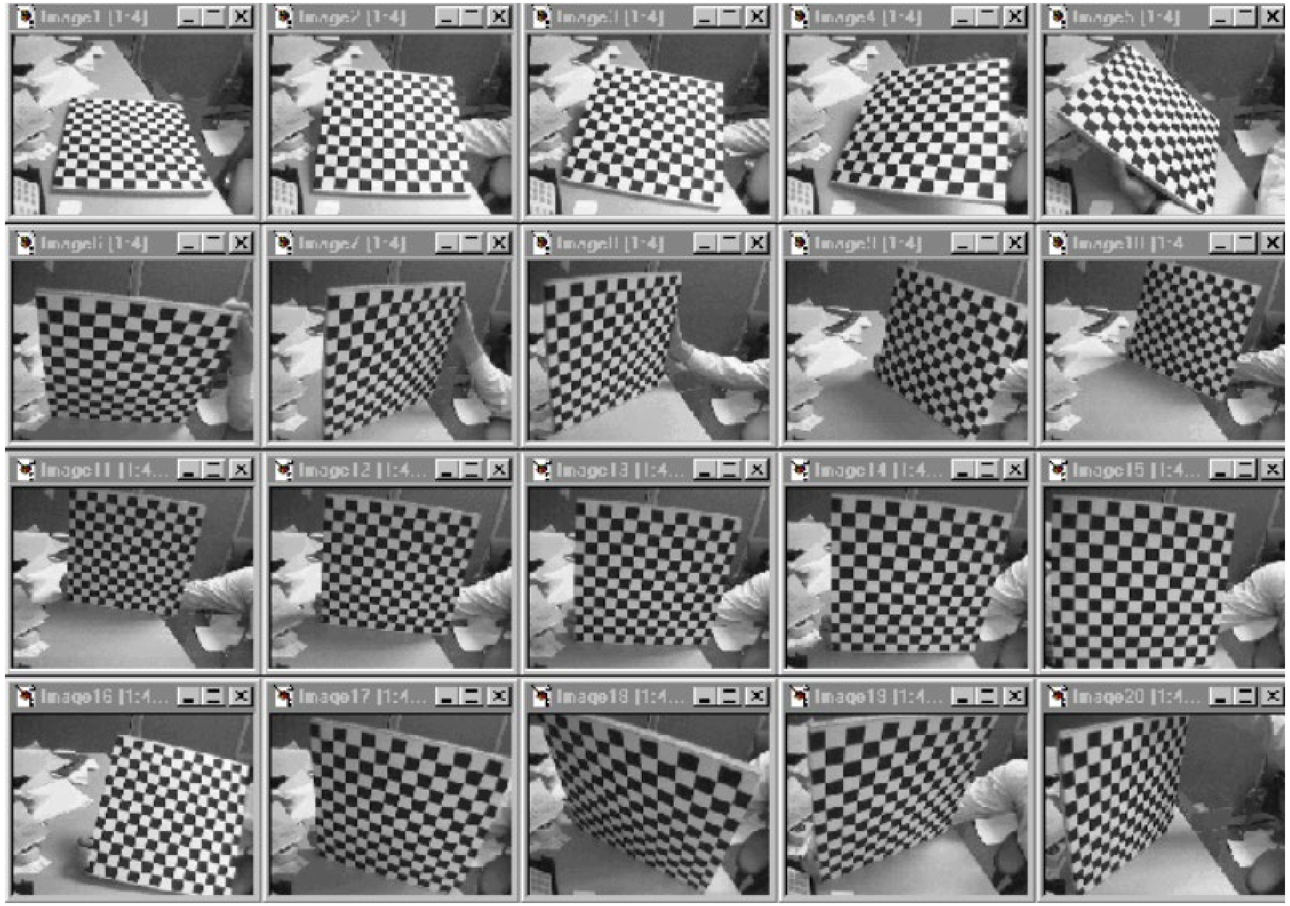


Image Source: Jean-Yves Bouguet

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

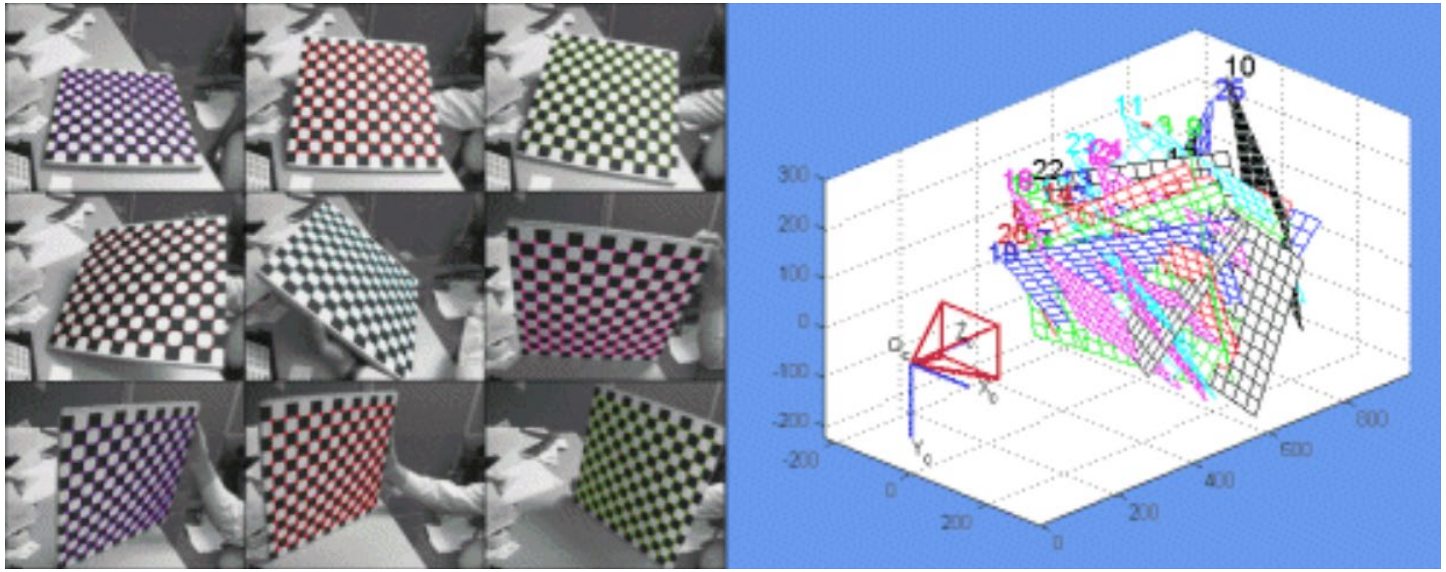


Image Source: Jean-Yves Bouguet

Code from my Research Lab:

Written by Diane Theriault

Published in

[Theriault et al., J Exp Biology, 2014]

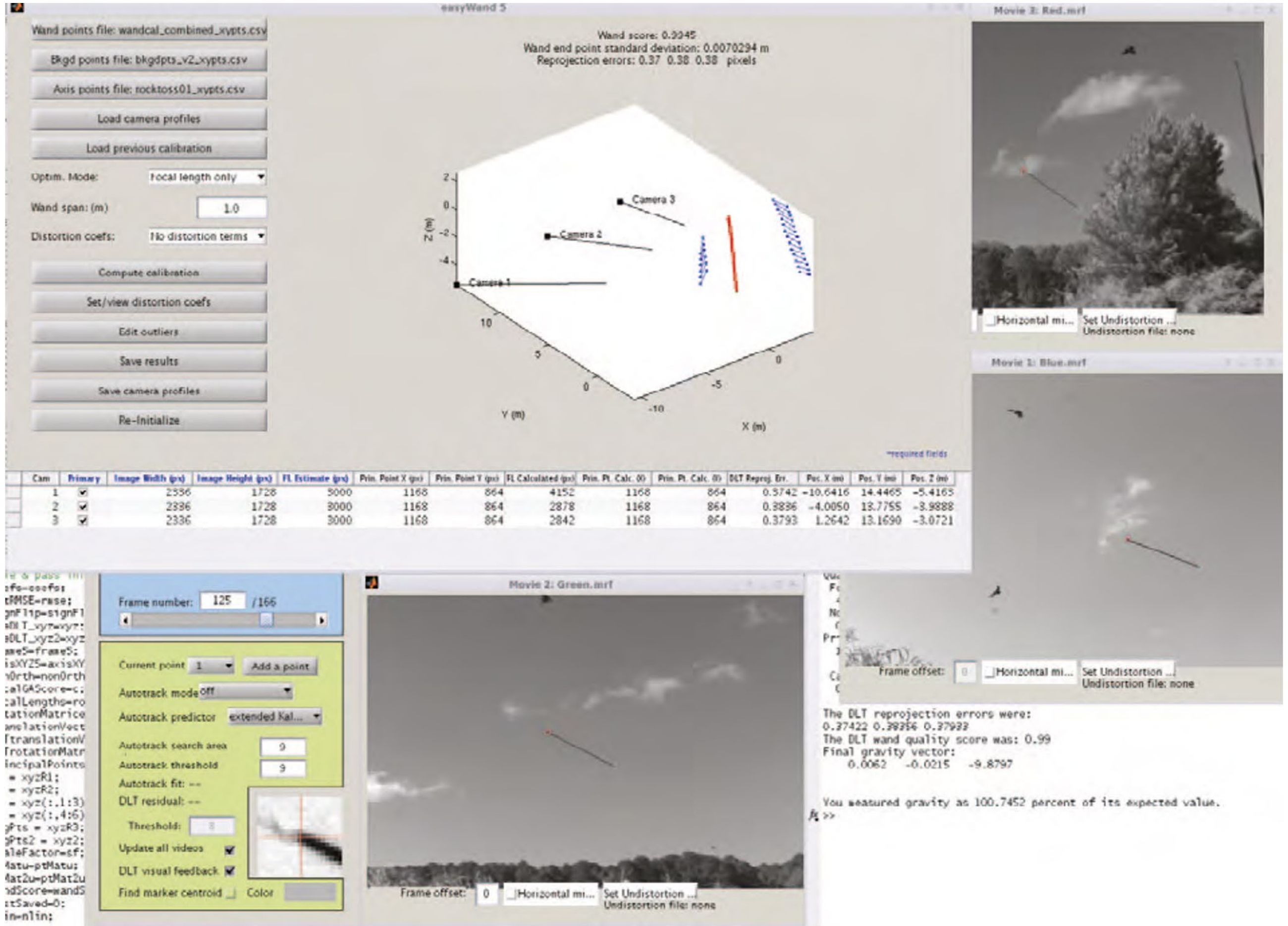
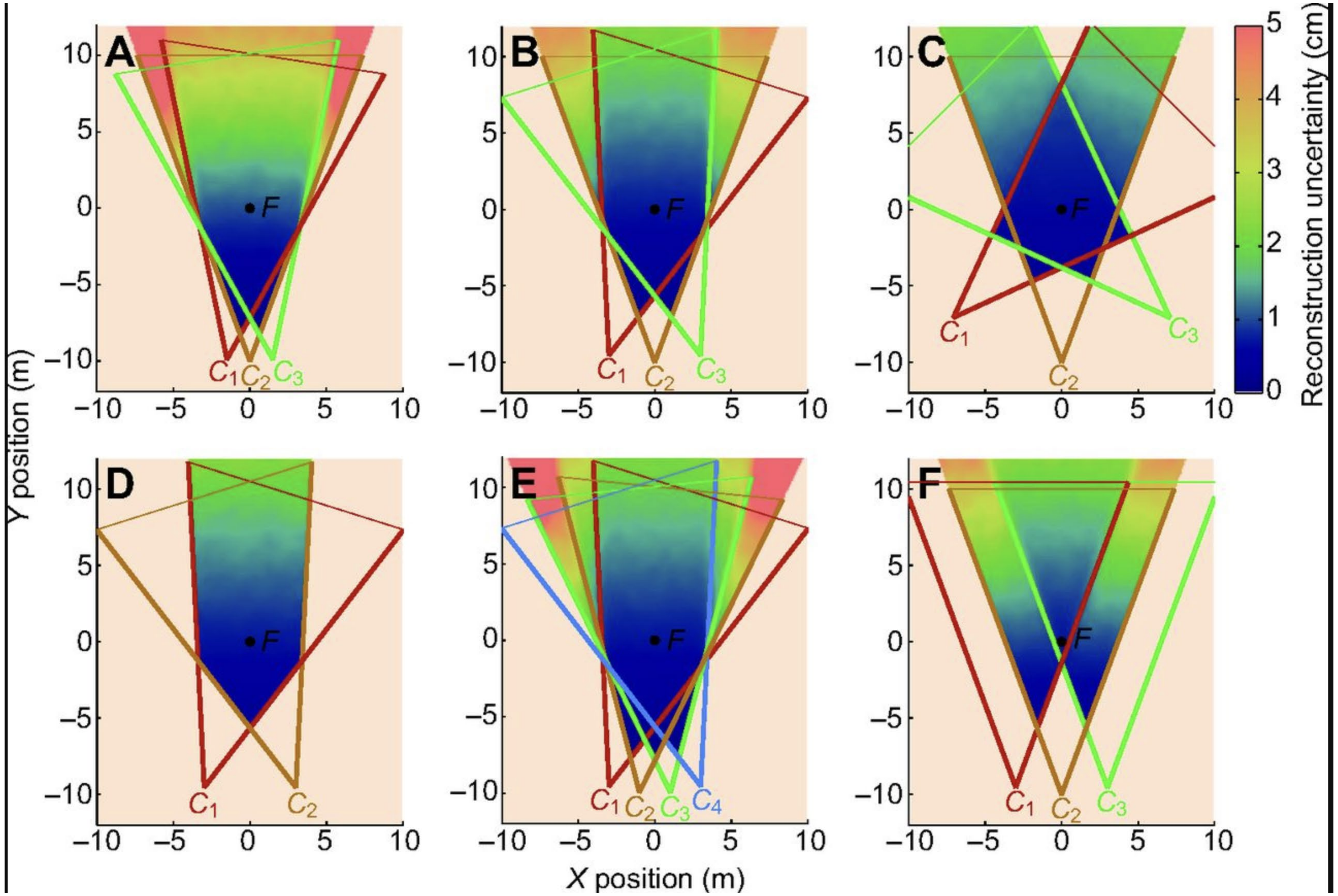


Figure S1:

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



Reconstruction uncertainty due to quantization effects is shown for six hypothetical camera configurations. The cameras were simulated to have a pixel width of 18 µm 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_{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).

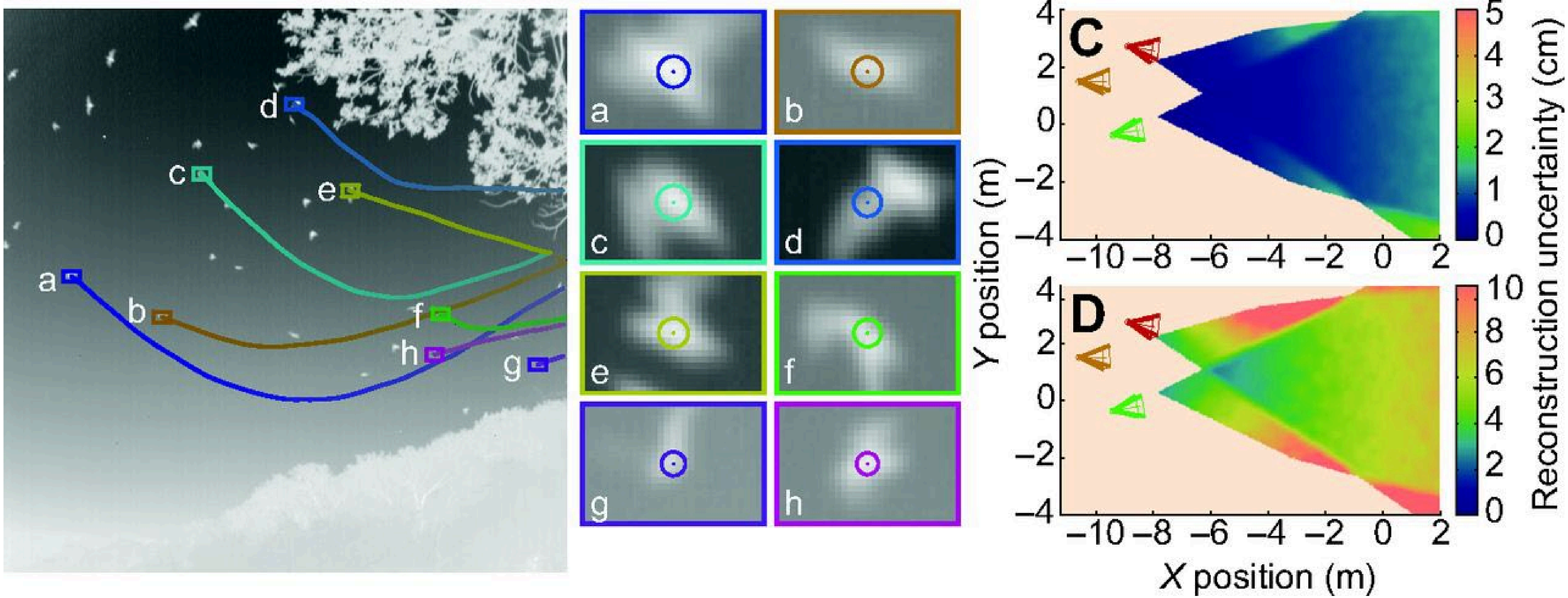
If the image planes of the cameras are parallel (F), the common view volume is smaller and further away from the cameras than in the other configurations.

These 2D cuts of the 3D view frustrums 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)

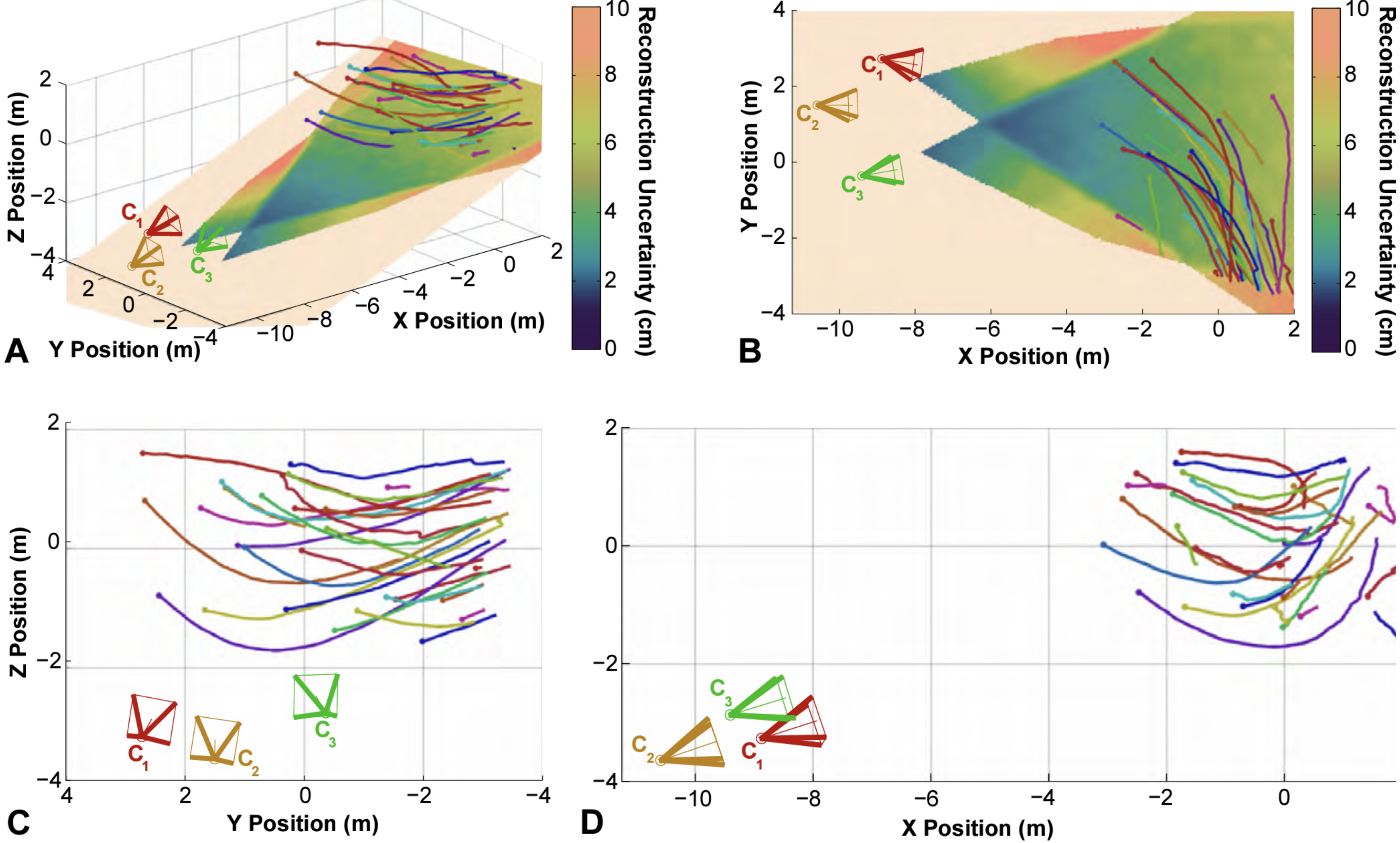


Figure S3: 3D flight trajectories of 28 Brazilian Free-tailed Bats during a 1-s interval are shown in the context of the spatially-varying reconstruction uncertainty arising due to both image quantization and image localization ambiguity from an oblique view (A) and from the top (B). The tracks are shown 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.

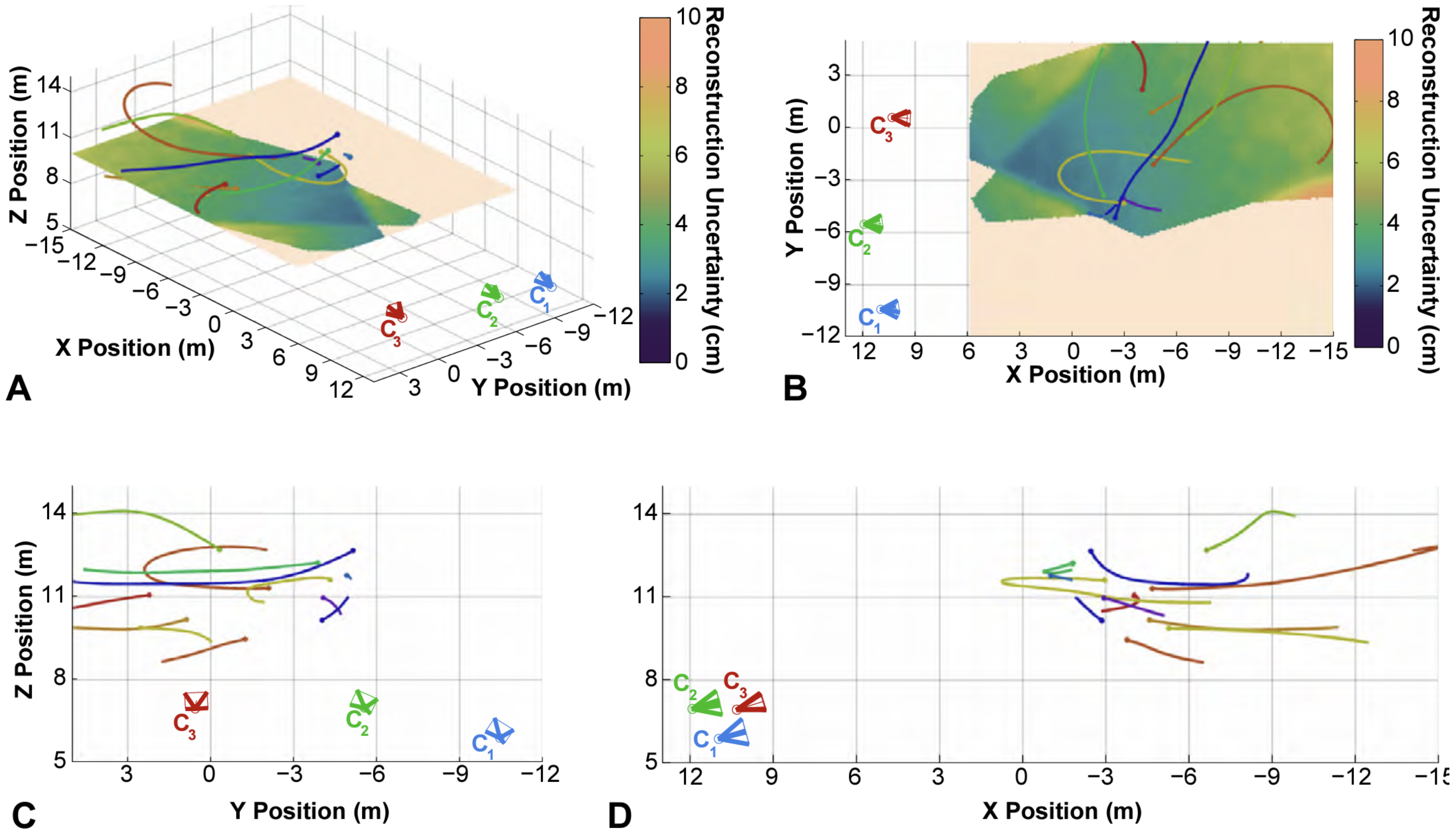
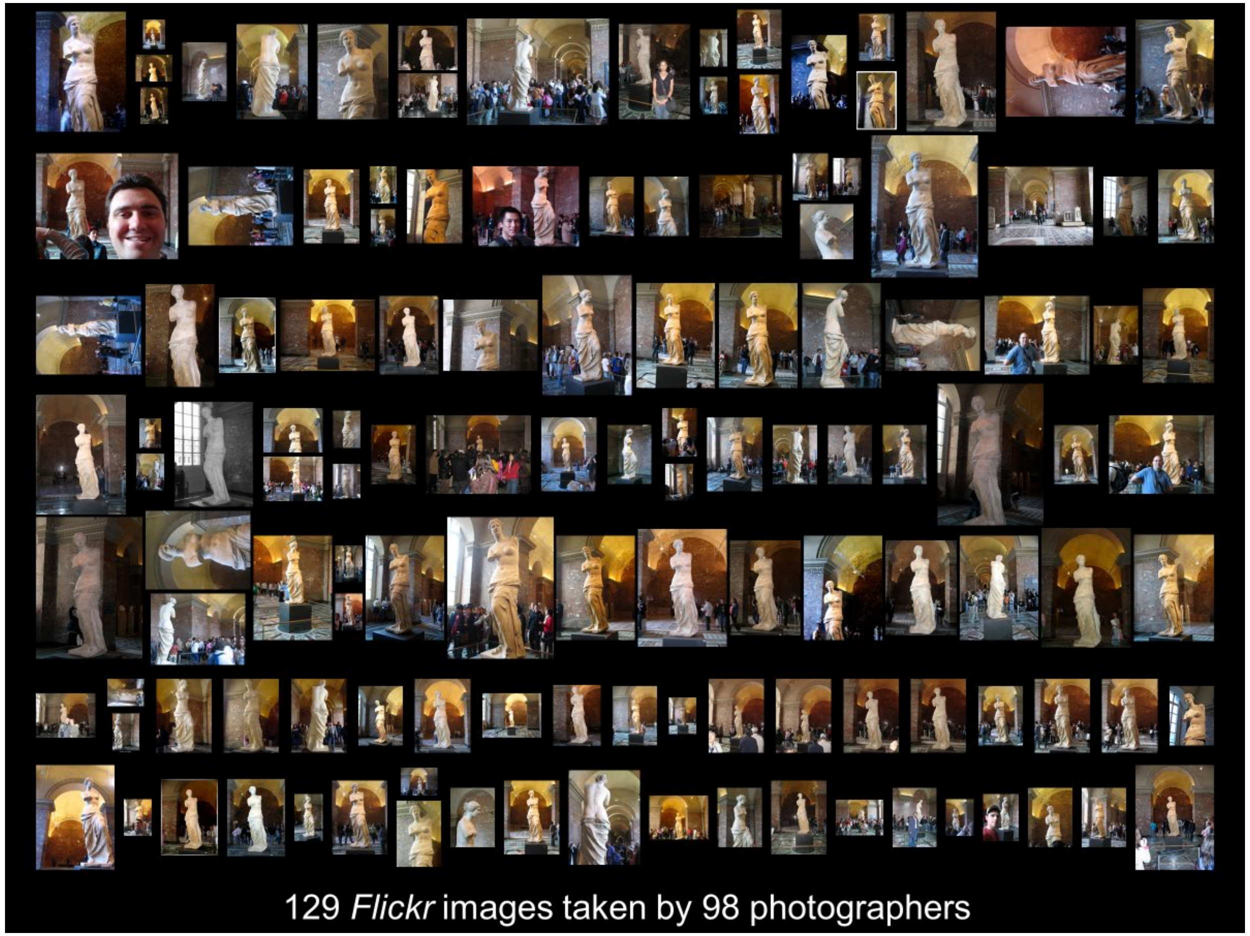


Figure S4: The flight paths of 12 Cliff Swallows during a 2.3-s interval are shown in the context of the spatially-varying reconstruction uncertainty arising due to both image quantization and image localization ambiguity from an oblique view (A) and from the top (B). The tracks are shown from the point 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](#)



First paper on Multiview Stereo and Using Internet Photo Collections to Reconstruct Scenes

Goesele, Snavely, et al., ICCV 2007, Snavely PhD thesis 2008

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

$$\min \sum_{i \in \mathbf{Cameras}} \sum_{j \in \mathbf{Points}} \left\| \mathbf{r}^{(j)}_{\text{camera,i}} - \pi_i\left(\mathbf{r}^{(j)}_{\text{camera,i}}\right) \right\|^2$$

where π_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

Schonberger & Frahm, CVPR, 2016: Structure-from-Motion Revisited

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

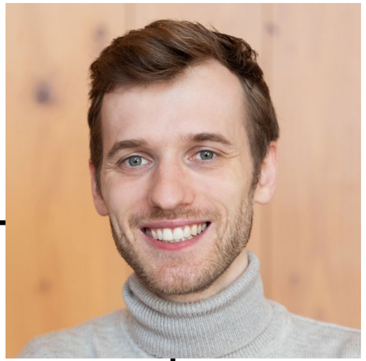
Output: 3D reconstruction of scene or object

Rome dataset

74,394 images

Schonberger & Frahm, CVPR 2016

MIT's 6.8300/6.8301 Advances in Computer Vision: Vincent Sitzmann's Lection on Multi-view Geometry in Spring 2023



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

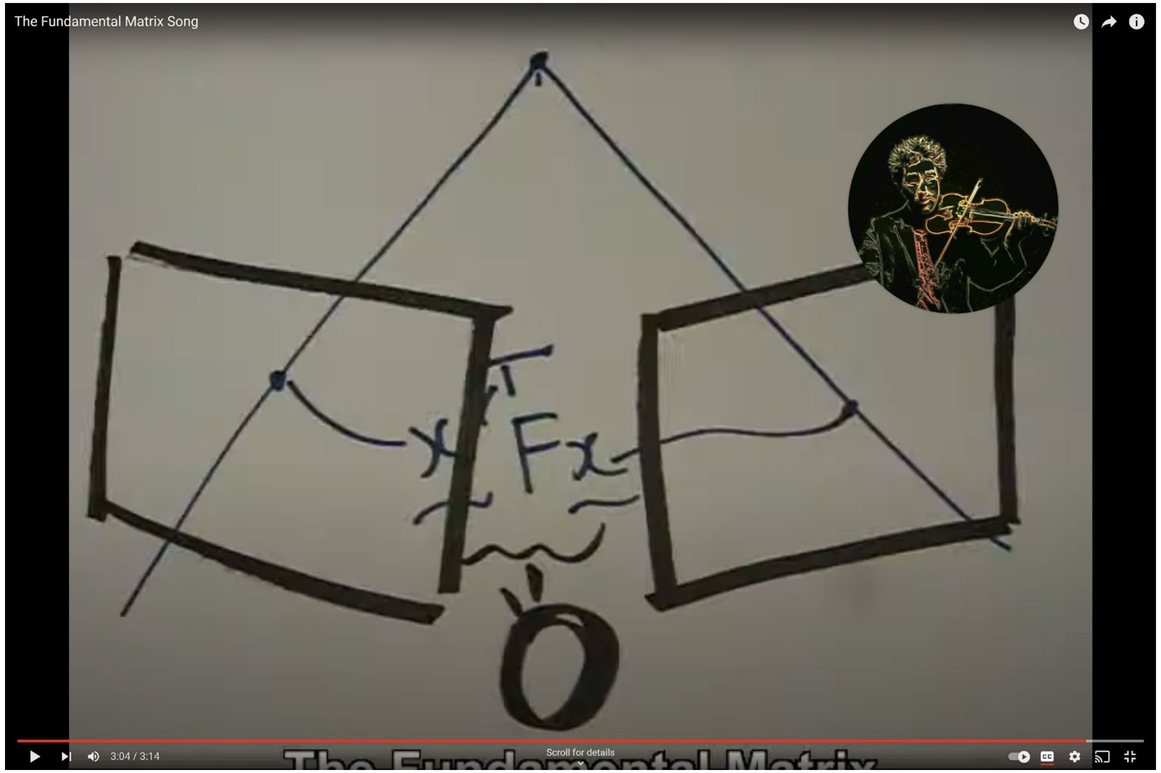
Parallel Tracking and Mapping for Small AR Workspaces

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