CS 598 3D Vision
Two-View Geometry

Shenlong Wang
UIUC

ILLINOIS

Some materials borrowed from Matthew O'Toole, Kris Kitani, Lana Lazebnik, Derek Hoeim, Sanja Fidler

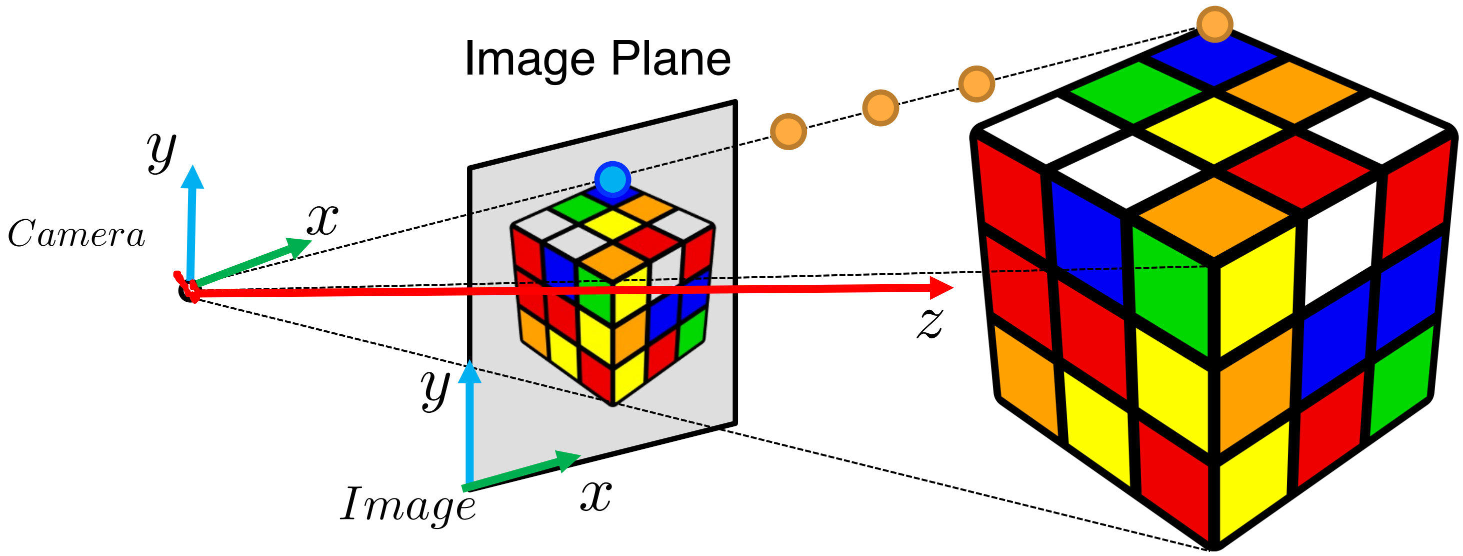
Logistics

- Quiz-1 is due!
- Thank you for sending out the survey!
- Group assignment will be out tonight / tomorrow morning.
- Next Tue is multi-view geometry (SFM and MVS);
- Next Thursday – role-play example run (Shenlong, Zhi-Hao and Albert)

Today's Agenda

- Triangulation
- Epipolar Geometry
- Stereo Matching
- Structured Light Cameras

Recap



Motivation: Multiple Views give 3D!

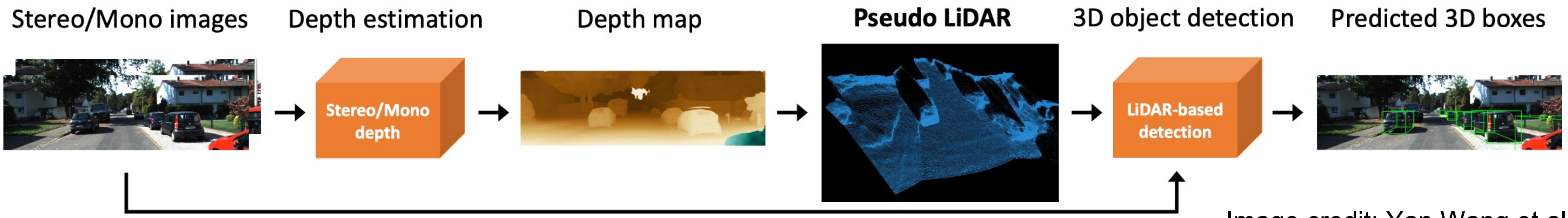
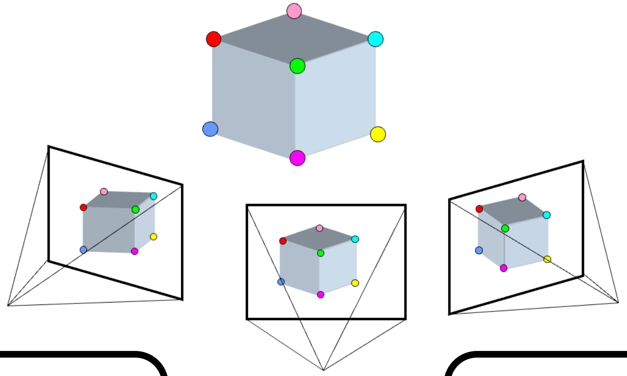


Image credit: Yan Wang et al.

Image credit: Jianxiong Xiao

Big picture: 3 key components in 3D

3D Points
(Structure)

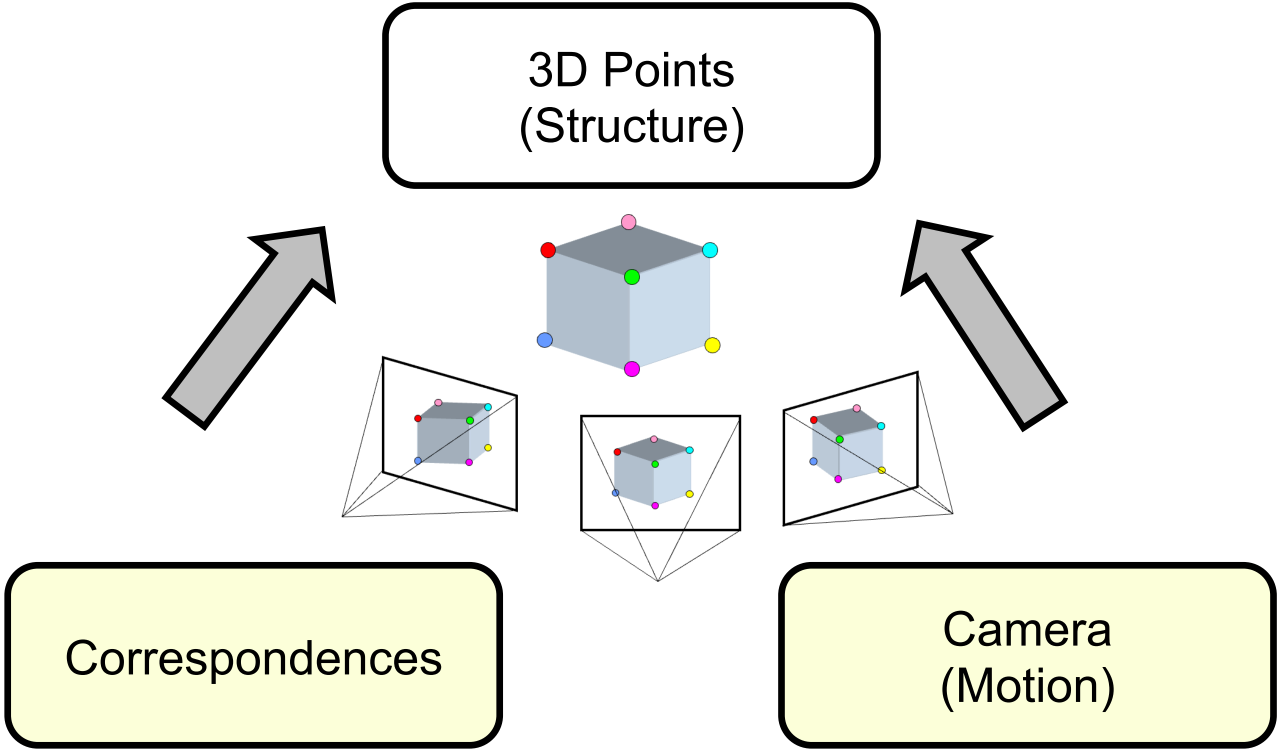


Correspondences

Camera
(Motion)

Angjoo Kanazawa

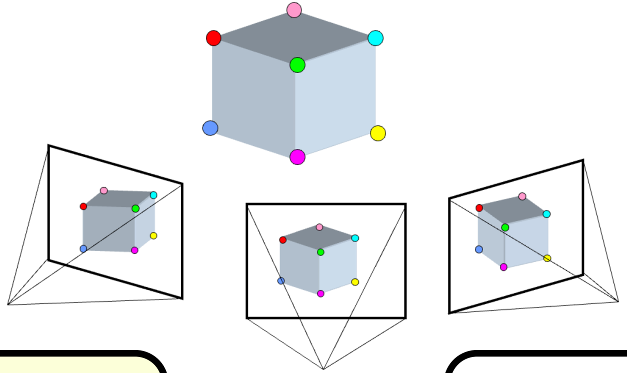
Big picture: 3 key components in 3D



Angjoo Kanazawa

Big picture: 3 key components in 3D

3D Points
(Structure)

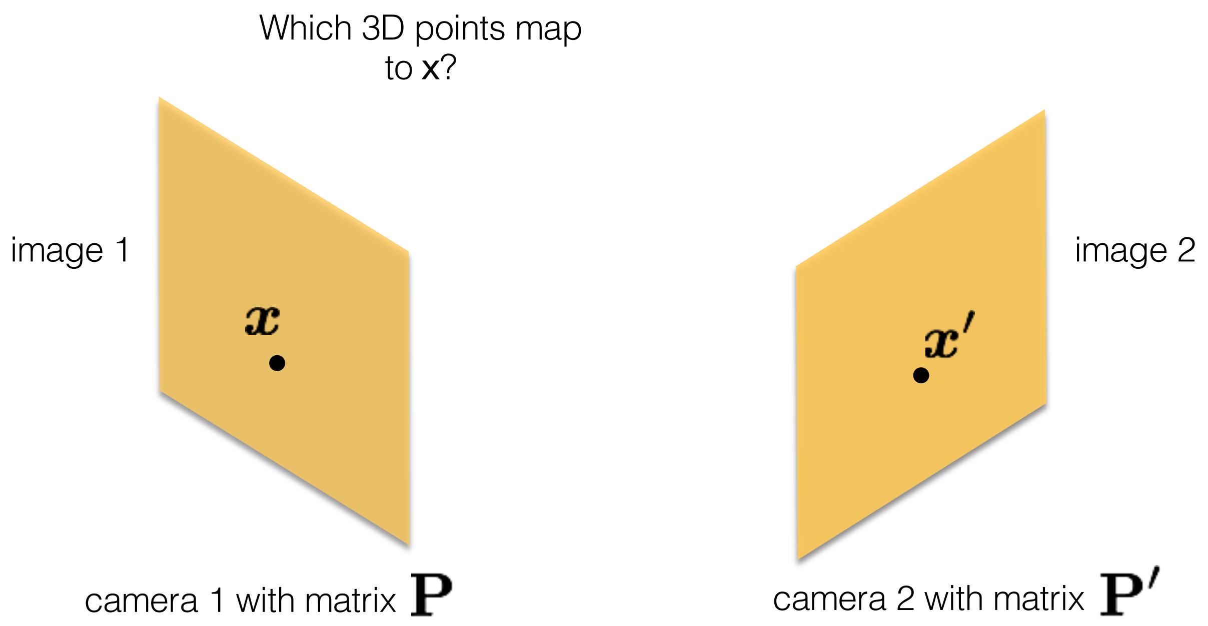


Correspondences

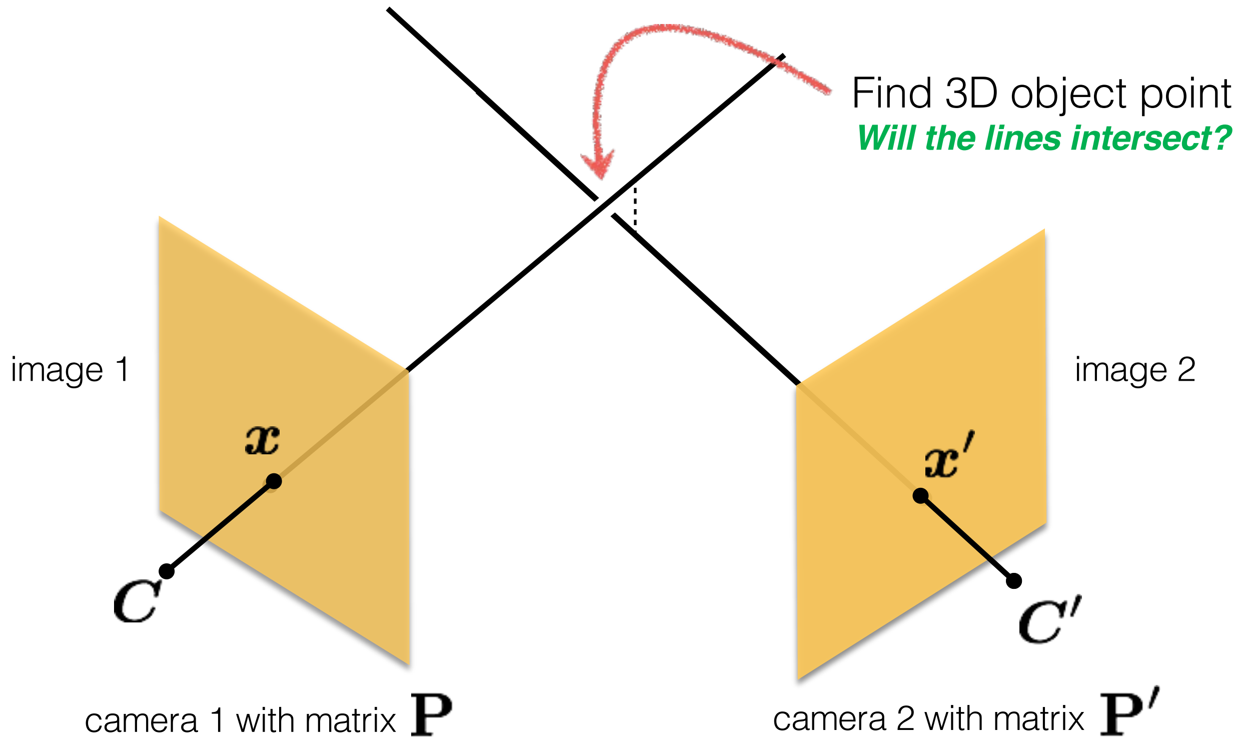
Camera
(Motion)

Angjoo Kanazawa

Triangulation



Triangulation



Triangulation

Given a set of (noisy) matched points

$$\{\boldsymbol{x}_i, \boldsymbol{x}'_i\}$$

and camera matrices

$$\mathbf{P}, \mathbf{P}'$$

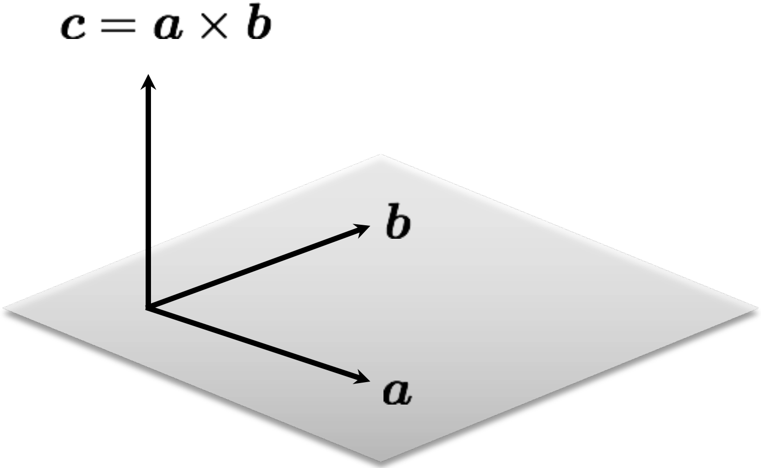
Estimate the 3D point

$$\mathbf{X}$$

Linear algebra reminder: cross product

Vector (cross) product

takes two vectors and returns a vector <u>perpendicular</u> to both



$$\boldsymbol{c} \cdot \boldsymbol{a} = 0 \qquad \boldsymbol{c} \cdot \boldsymbol{b} = 0$$

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

cross product of two vectors in the same direction is zero vector

$$\boldsymbol{a} \times \boldsymbol{a} = 0$$

remember this!!!

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

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

Do the same after first expanding out the camera matrix and points

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} \text{---} & \boldsymbol{p}_1^\top & \text{---} \\ \text{---} & \boldsymbol{p}_2^\top & \text{---} \\ \text{---} & \boldsymbol{p}_3^\top & \text{---} \end{bmatrix} \begin{bmatrix} | \\ \boldsymbol{X} \\ | \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} \boldsymbol{p}_1^\top \boldsymbol{X} \\ \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_3^\top \boldsymbol{X} \end{bmatrix}$$

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

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} \text{---} & \boldsymbol{p}_1^\top & \text{---} \\ \text{---} & \boldsymbol{p}_2^\top & \text{---} \\ \text{---} & \boldsymbol{p}_3^\top & \text{---} \end{bmatrix} \begin{bmatrix} | \\ \boldsymbol{X} \\ | \end{bmatrix}$$

Do the same after first expanding out the camera matrix and points

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} \boldsymbol{p}_1^\top \boldsymbol{X} \\ \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_3^\top \boldsymbol{X} \end{bmatrix}$$

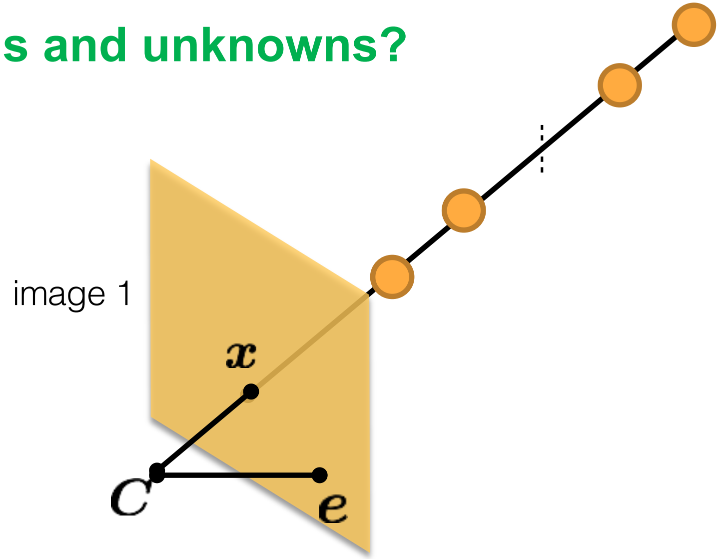
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \times \begin{bmatrix} \boldsymbol{p}_1^\top \boldsymbol{X} \\ \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_3^\top \boldsymbol{X} \end{bmatrix} = \begin{bmatrix} y\boldsymbol{p}_3^\top \boldsymbol{X} - \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_1^\top \boldsymbol{X} - x\boldsymbol{p}_3^\top \boldsymbol{X} \\ x\boldsymbol{p}_2^\top \boldsymbol{X} - y\boldsymbol{p}_1^\top \boldsymbol{X} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Using the fact that the cross product should be zero

$$\mathbf{x} \times \mathbf{P}\boldsymbol{X} = \mathbf{0}$$

Knowns and unknowns?

$$\left[\begin{array}{c} y\boldsymbol{p}_3^\top \boldsymbol{X} - \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_1^\top \boldsymbol{X} - x\boldsymbol{p}_3^\top \boldsymbol{X} \\ x\boldsymbol{p}_2^\top \boldsymbol{X} - y\boldsymbol{p}_1^\top \boldsymbol{X} \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ 0 \end{array}\right]$$



Third line is a linear combination of the first and second lines.
(x times the first line plus y times the second line)

One 2D to 3D point correspondence give you 2 equations

Remove third row, and rearrange as system on unknowns

$$\begin{bmatrix} y\boldsymbol{p}_3^\top - \boldsymbol{p}_2^\top \\ \boldsymbol{p}_1^\top - x\boldsymbol{p}_3^\top \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\mathbf{A}_i \boldsymbol{X} = \mathbf{0}$$

Now we can make a system of linear equations (two lines for each 2D point correspondence)

Concatenate the 2D points from both images

Two rows from camera one

Two rows from camera two

$$\begin{bmatrix} y\boldsymbol{p}_3^\top - \boldsymbol{p}_2^\top \\ \boldsymbol{p}_1^\top - x\boldsymbol{p}_3^\top \\ y'\boldsymbol{p}_3'^\top - \boldsymbol{p}_2'^\top \\ \boldsymbol{p}_1'^\top - x'\boldsymbol{p}_3'^\top \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

sanity check! dimensions?

$$\mathbf{A}\boldsymbol{X} = \mathbf{0}$$

How do we solve homogeneous linear system?

Concatenate the 2D points from both images

$$\begin{bmatrix} y\boldsymbol{p}_3^\top - \boldsymbol{p}_2^\top \\ \boldsymbol{p}_1^\top - x\boldsymbol{p}_3^\top \\ y'\boldsymbol{p}_3'^\top - \boldsymbol{p}_2'^\top \\ \boldsymbol{p}_1'^\top - x'\boldsymbol{p}_3'^\top \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{A}\boldsymbol{X} = \mathbf{0}$$

How do we solve homogeneous linear system?

S V D ! (or Gaussian Elimination, QR, etc.…)

Triangulation

Input:

- Camera matrix
- A pair of correspondence of a point

Output:

- 3D location of the point

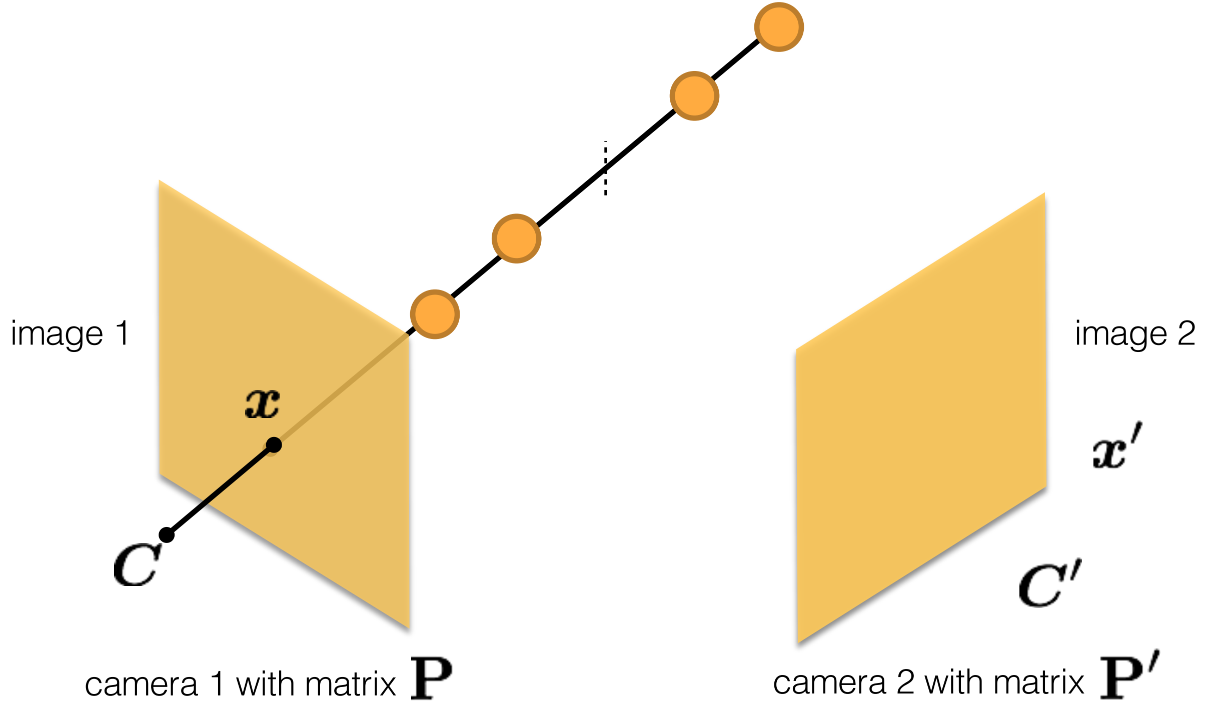
Today’s Agenda

- Triangulation
- **Epipolar Geometry**
- Stereo Matching
- Structured Light Cameras

Epipolar Geometry

Given a 2D point location from image 1

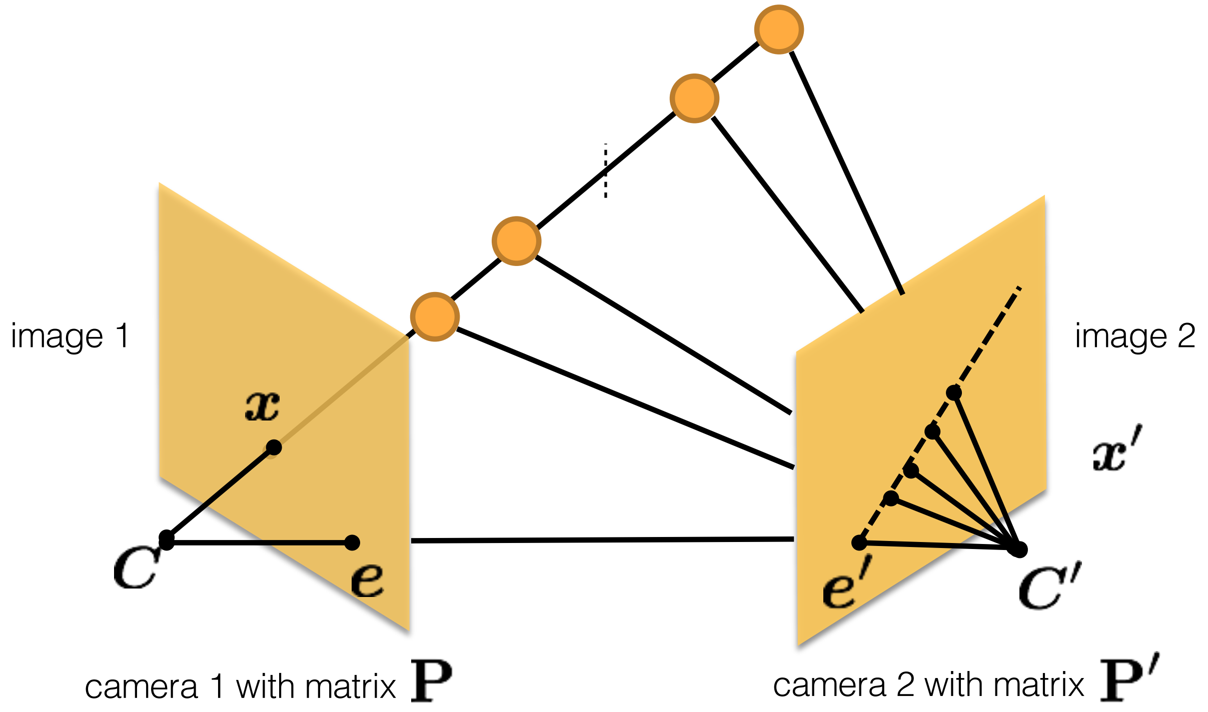
Where the correspondence could be at image 2?



Epipolar Geometry

Given a 2D point location from image 1

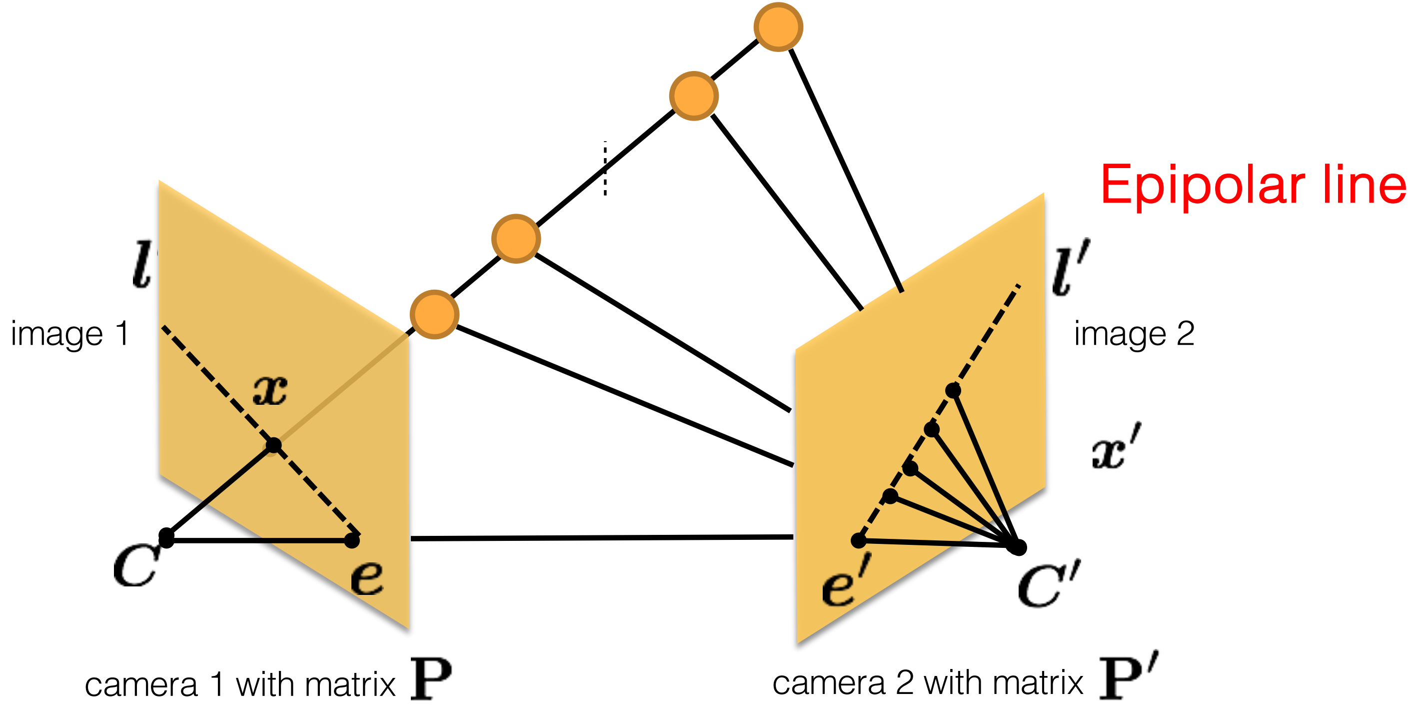
Where the correspondence could be at image 2?



Epipolar Geometry

Given a 2D point location from image 1

Where the correspondence could be at image 2?

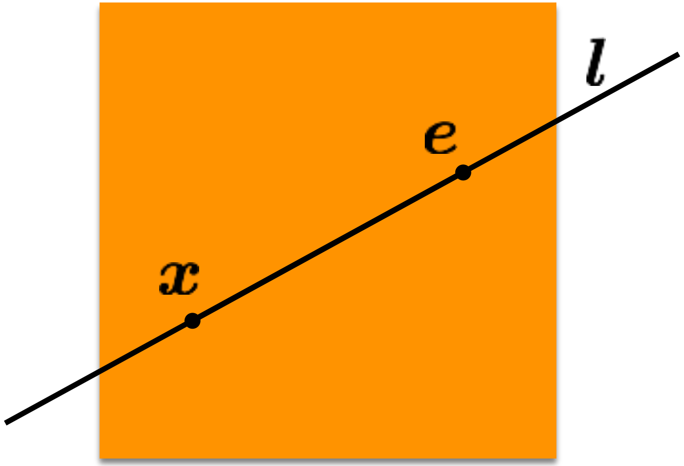


2D Line Equation

$$ax + by + c = 0$$

in vector form

$$\boldsymbol{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$



If the point $\boldsymbol{x}$ is on the epipolar line $\boldsymbol{l}$ then

$$\boldsymbol{x}^\top \boldsymbol{l} = 0$$

Epipolar Line

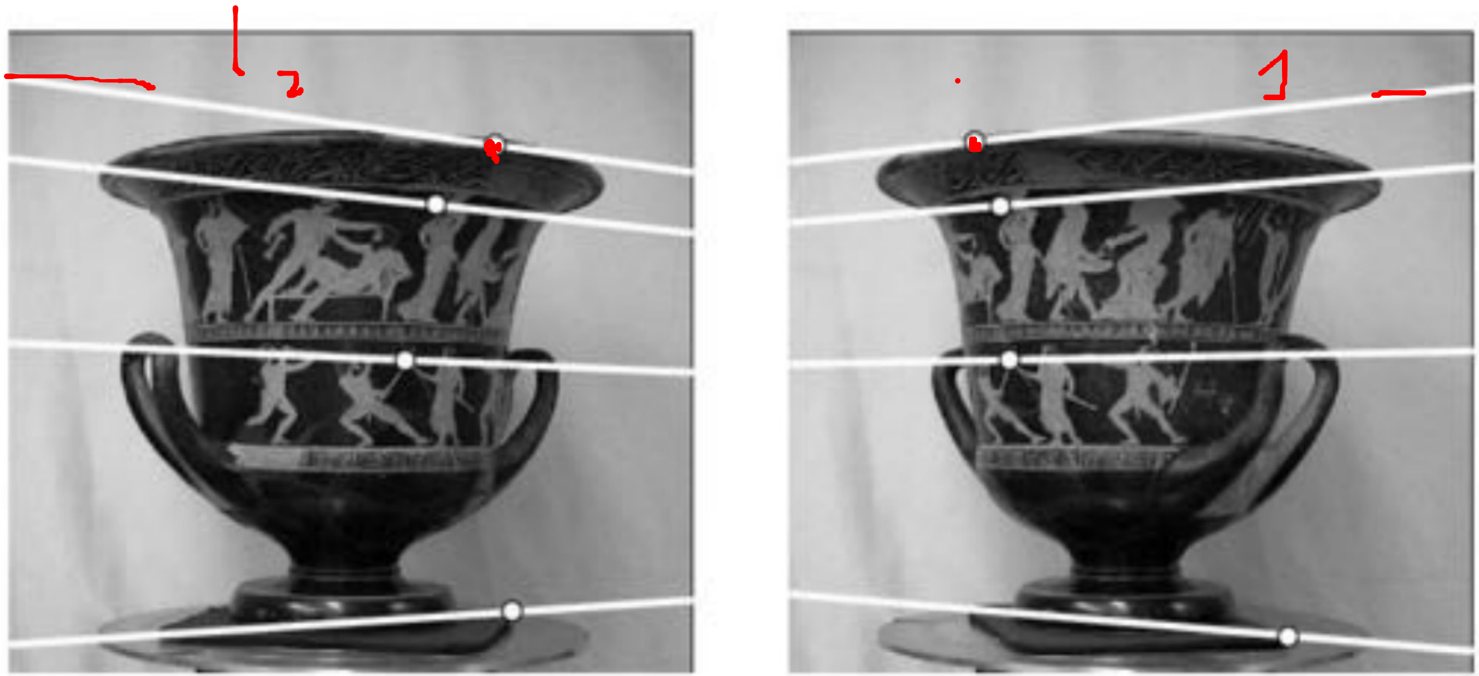
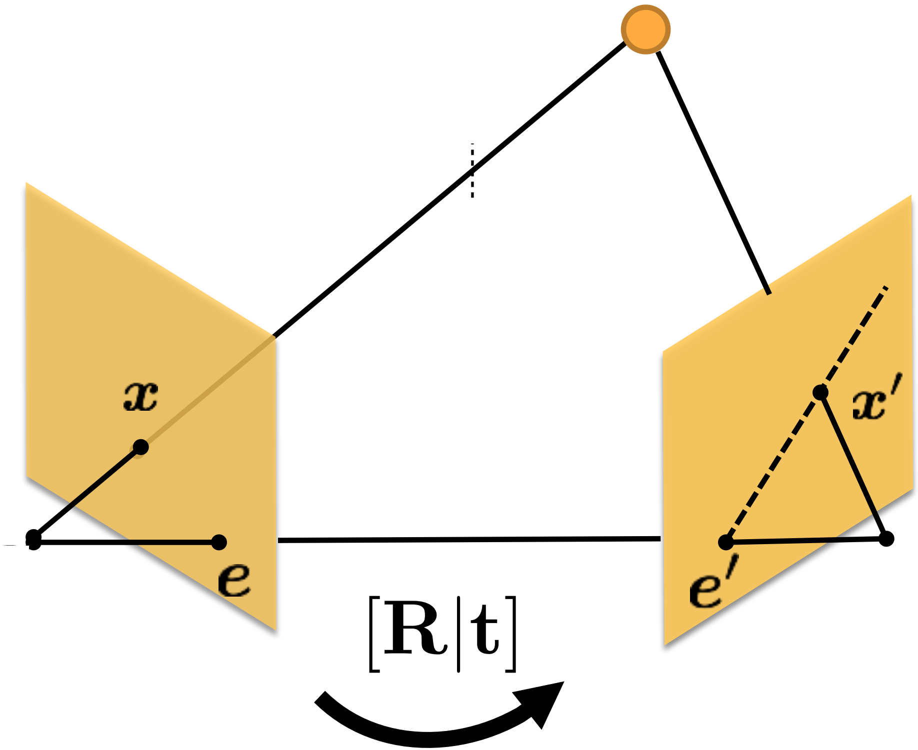


Image credit: Zisserman

Epipolar Geometry

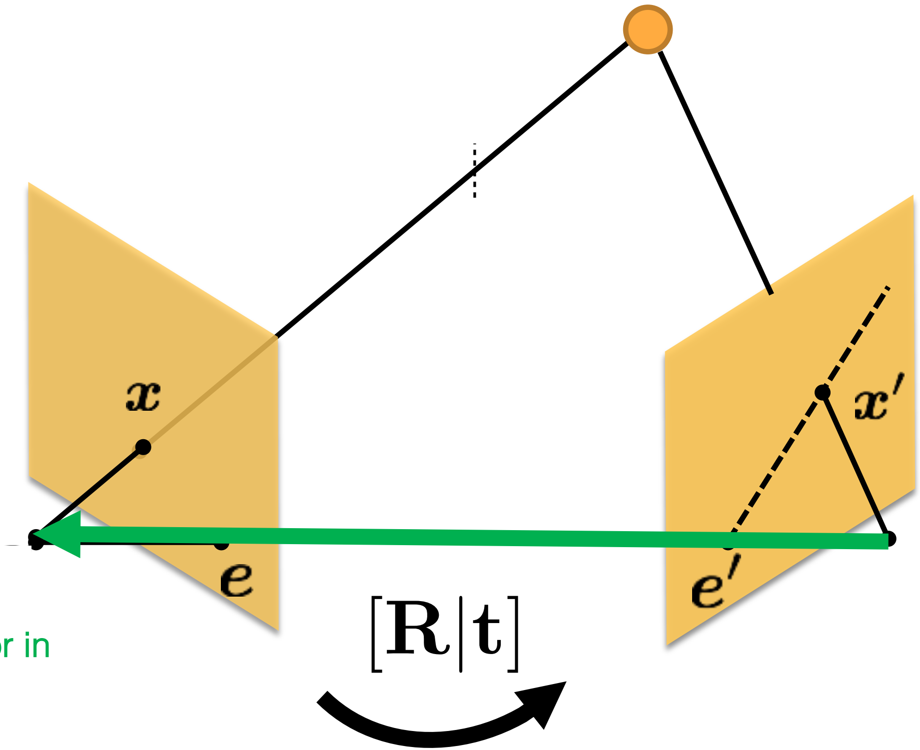
Correspondence location in camera1 frame



Correspondence location in camera2 frame

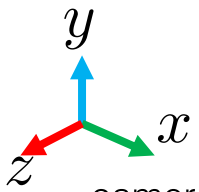
Epipolar Geometry

Correspondence location in camera1 frame

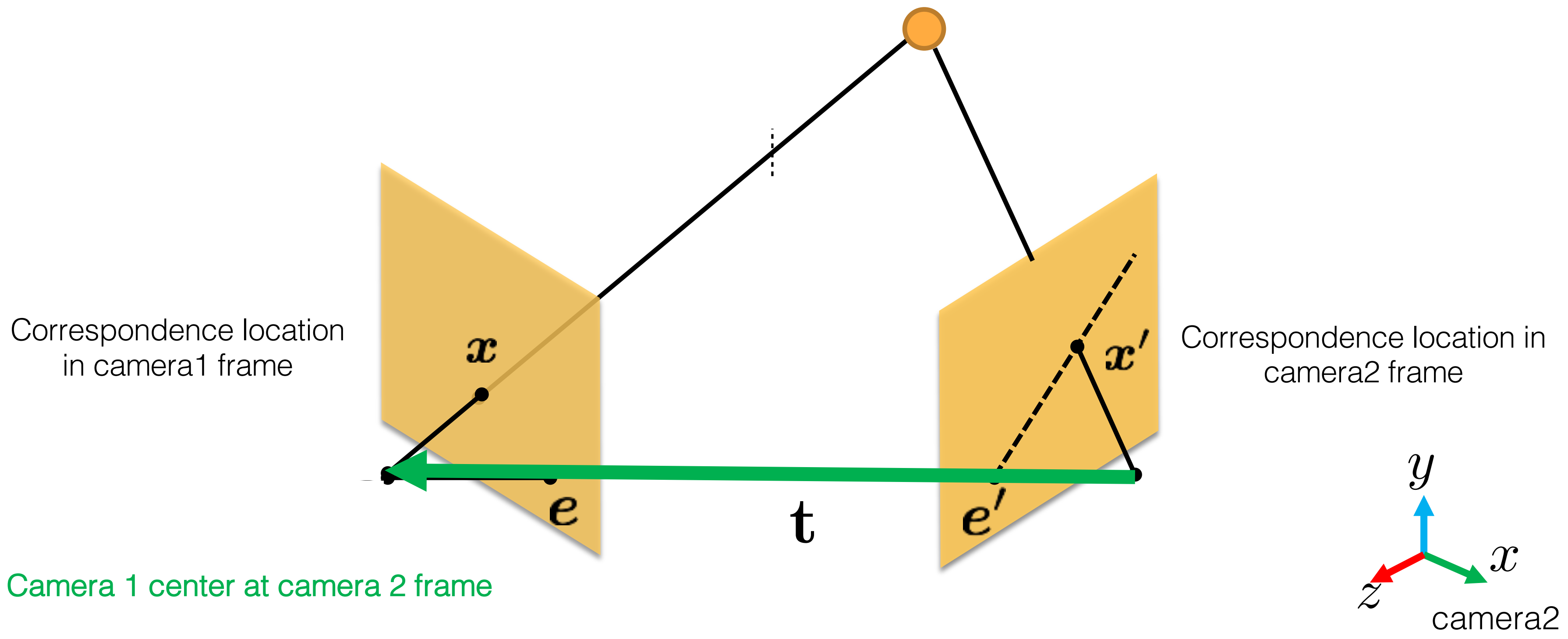


Correspondence location in camera2 frame

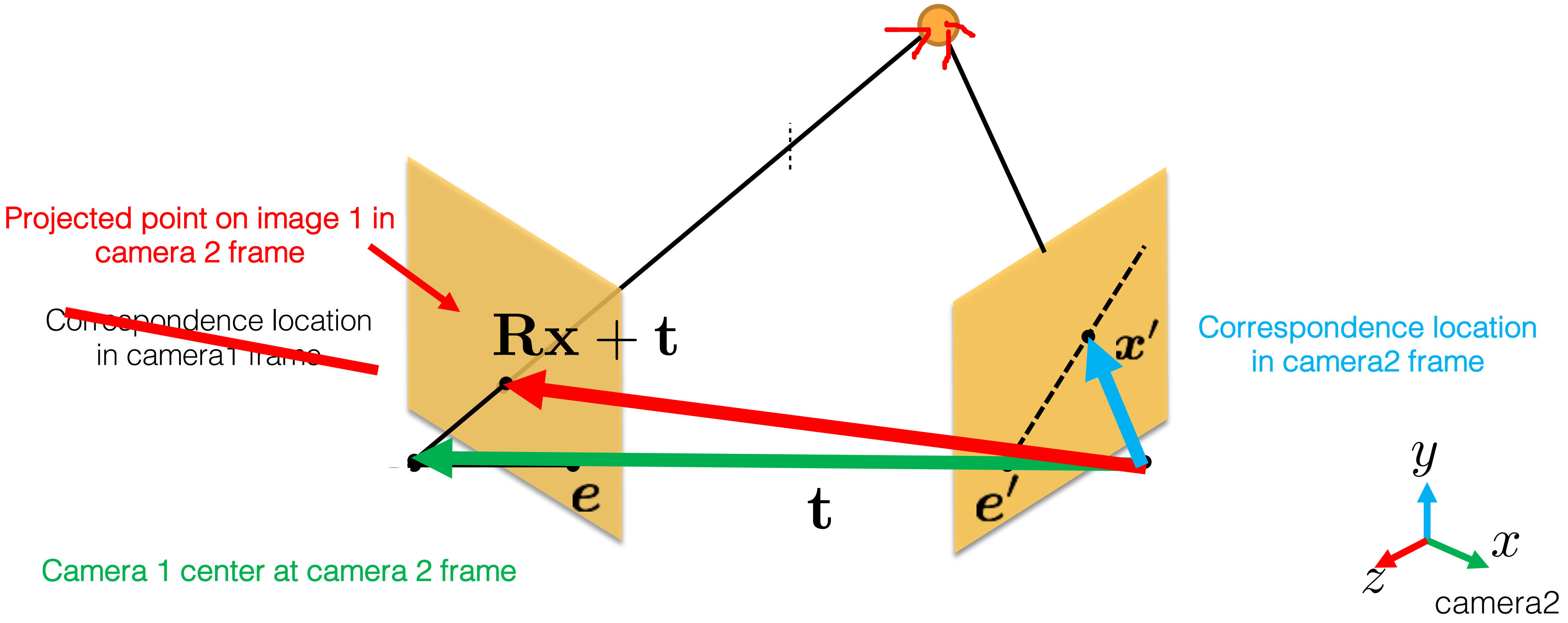
Could you write this vector in camera 2 frame?



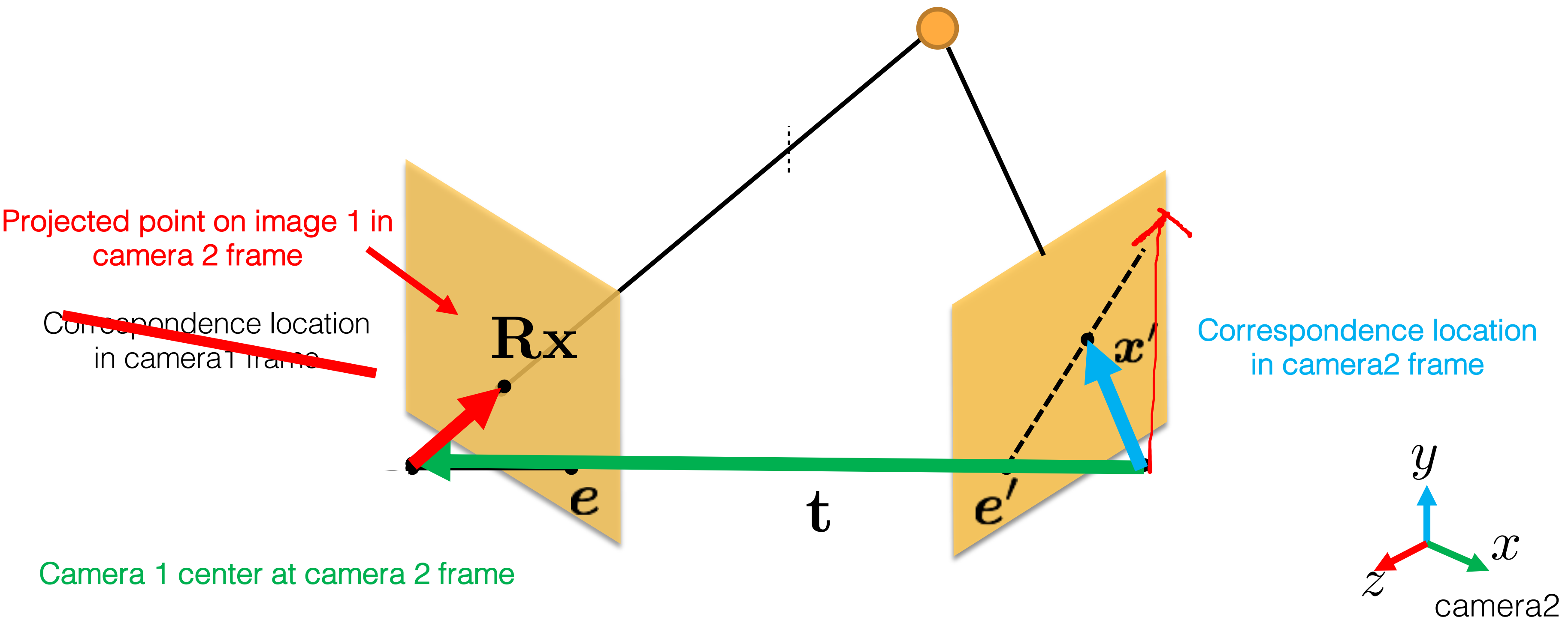
Epipolar Geometry



Epipolar Geometry



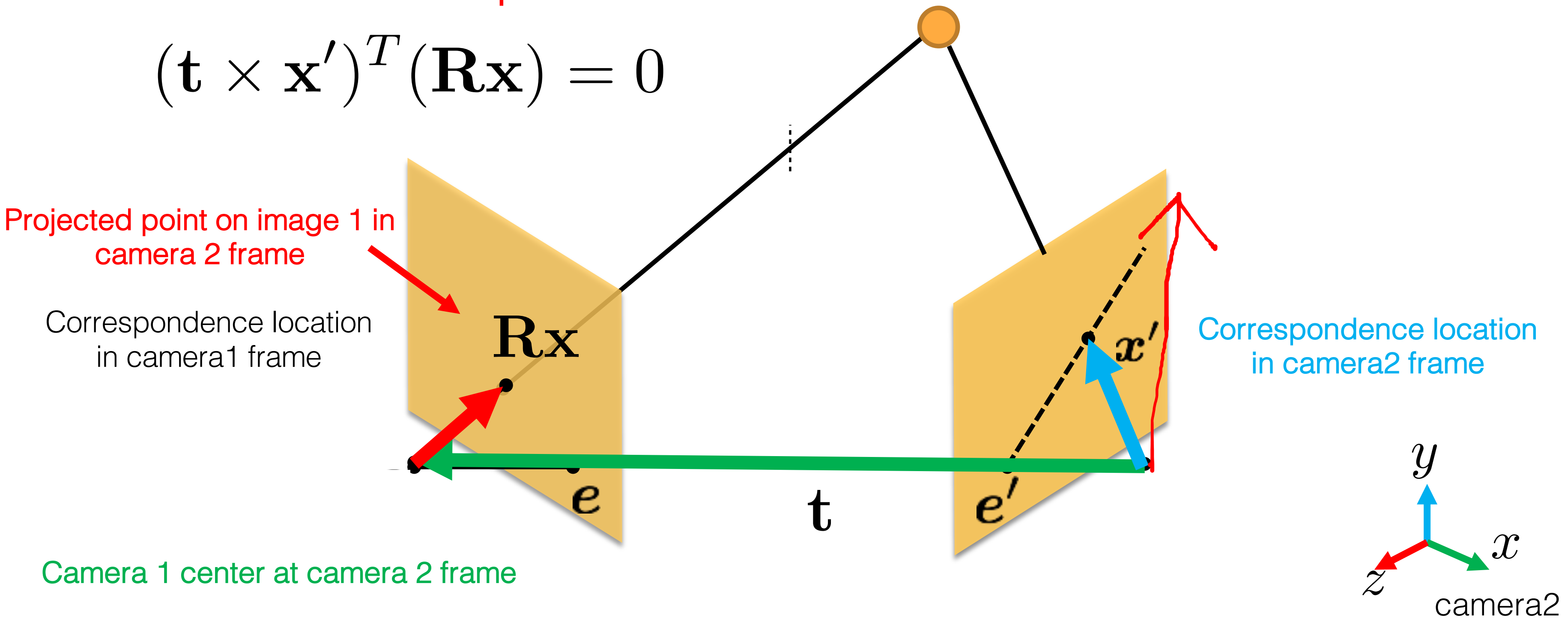
Epipolar Geometry



Epipolar Geometry

- Three vectors are coplanar:

$$(\mathbf{t} \times \mathbf{x}')^T (\mathbf{R}\mathbf{x}) = 0$$



Skew Symmetric Matrix

$$[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b}$$

transpose equals its negative

Essential Matrix

$$(\mathbf{t} \times \mathbf{x}')^T(\mathbf{R}\mathbf{x}) = 0$$

$$(\mathbf{x}')^T\mathbf{E}\mathbf{x} = 0, \mathbf{E} = [\mathbf{t}]_\times\mathbf{R}$$

Epipolar lines

$$\boldsymbol{x}^\top \boldsymbol{l} = 0 \qquad \boldsymbol{x}'^\top \boldsymbol{l}' = 0$$

$$\boldsymbol{l}' = \mathbf{E}\boldsymbol{x} \qquad \boldsymbol{l} = \mathbf{E}^T\boldsymbol{x}'$$

Essential Matrix

$$(\mathbf{t} \times \mathbf{x}')^T (\mathbf{R}\mathbf{x}) = 0$$

How many degree of freedom?

$$(\mathbf{x}')^T \mathbf{E}\mathbf{x} = 0, \mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$$

3 3

Epipolar lines

$$\boldsymbol{x}^\top \boldsymbol{l} = 0 \qquad \boldsymbol{x}'^\top \boldsymbol{l}' = 0$$

$$\boldsymbol{l}' = \mathbf{E}\boldsymbol{x} \qquad \boldsymbol{l} = \mathbf{E}^T \boldsymbol{x}'$$

Essential Matrix

$$(\mathbf{t} \times \mathbf{x}')^T(\mathbf{R}\mathbf{x}) = 0$$

How many degree of freedom?

$$(\mathbf{x}')^T\mathbf{E}\mathbf{x} = 0, \mathbf{E} = [\mathbf{t}]_\times\mathbf{R}$$

6 DOF rigid – 1 DOF scale ambiguity

Epipolar lines

$$\boldsymbol{x}^\top \boldsymbol{l} = 0 \qquad \boldsymbol{x}'^\top \boldsymbol{l}' = 0$$

$$\boldsymbol{l}' = \mathbf{E}\boldsymbol{x} \qquad \boldsymbol{l} = \mathbf{E}^T\boldsymbol{x}'$$

Essential Matrix

$$(\mathbf{x}')^T \mathbf{E}\mathbf{x} = 0, \mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$$

Normalized camera coordinate: everything in camera coordinate but z is normalized to 1

The fundamental matrix song: http://danielwedge.com/fmatrix/

Fundamental Matrix

$$(\mathbf{x}')^T \mathbf{E} \mathbf{x} = 0, \mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$$

Normalized camera coordinate: everything in camera coordinate but z is normalized to 1

$$((\mathbf{K}')^{-1}\mathbf{x}')^T \mathbf{E} (\mathbf{K}^{-1}\mathbf{x}) = 0$$

Image coordinate

$$(\mathbf{x}')^T \mathbf{F} \mathbf{x} = 0 \qquad \mathbf{E} = (\mathbf{K}')^\top \, \mathbf{F} \, \mathbf{K}$$

The fundamental matrix song: http://danielwedge.com/fmatrix/

Camera Models

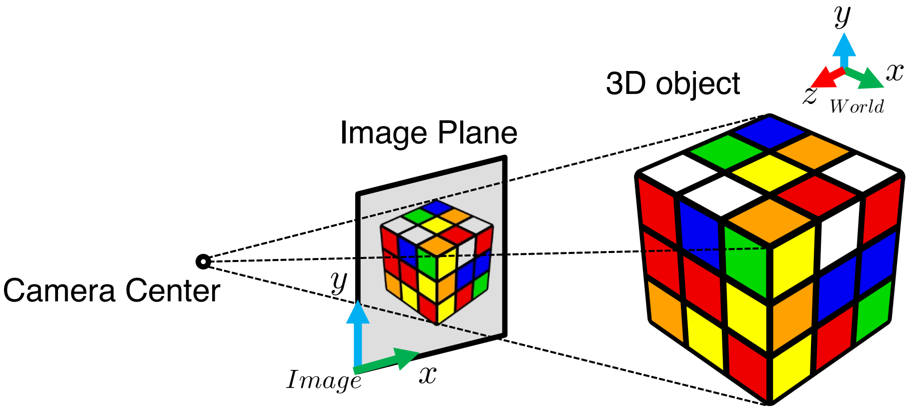
A camera is a mapping from the 3D world to a 2D image

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

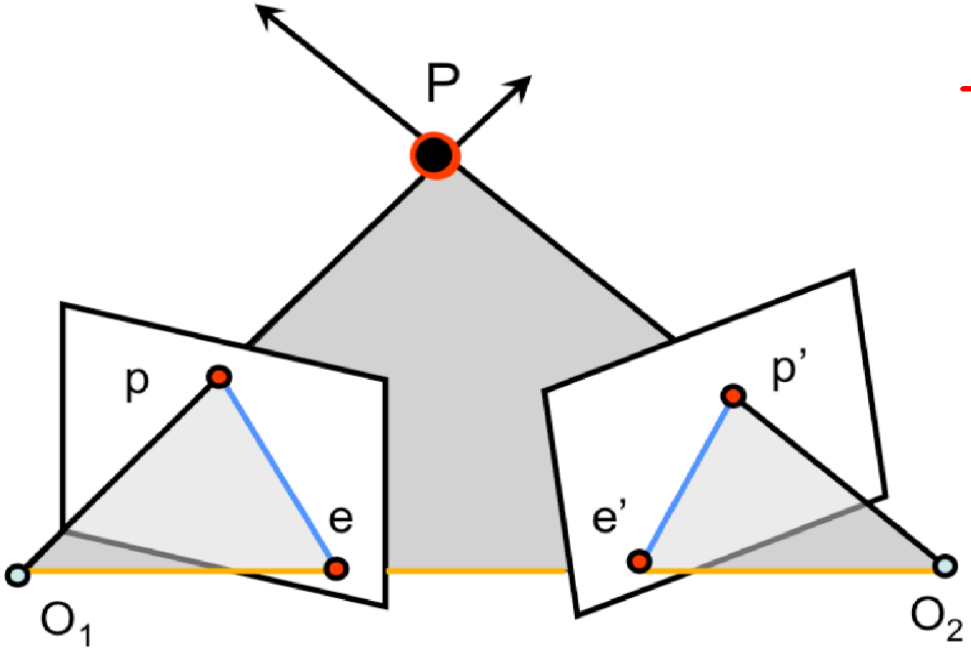
$$\mathbf{P} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{array}\right]$$

intrinsic parameters

extrinsic parameters



Epipolar Geometry



$$(\mathbf{x}')^T\mathbf{E}\mathbf{x} = 0, \mathbf{E} = [\mathbf{t}]_\times \mathbf{R}$$

$$\begin{bmatrix} y\boldsymbol{p}_3^\top - \boldsymbol{p}_2^\top \\ \boldsymbol{p}_1^\top - x\boldsymbol{p}_3^\top \\ y'\boldsymbol{p}_3'^\top - \boldsymbol{p}_2'^\top \\ \boldsymbol{p}_1'^\top - x'\boldsymbol{p}_3'^\top \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Today’s Agenda

- Triangulation
- Epipolar Geometry
- **Stereo Matching**
- Structured Light Cameras

If Cameras are Frontal Parallel..

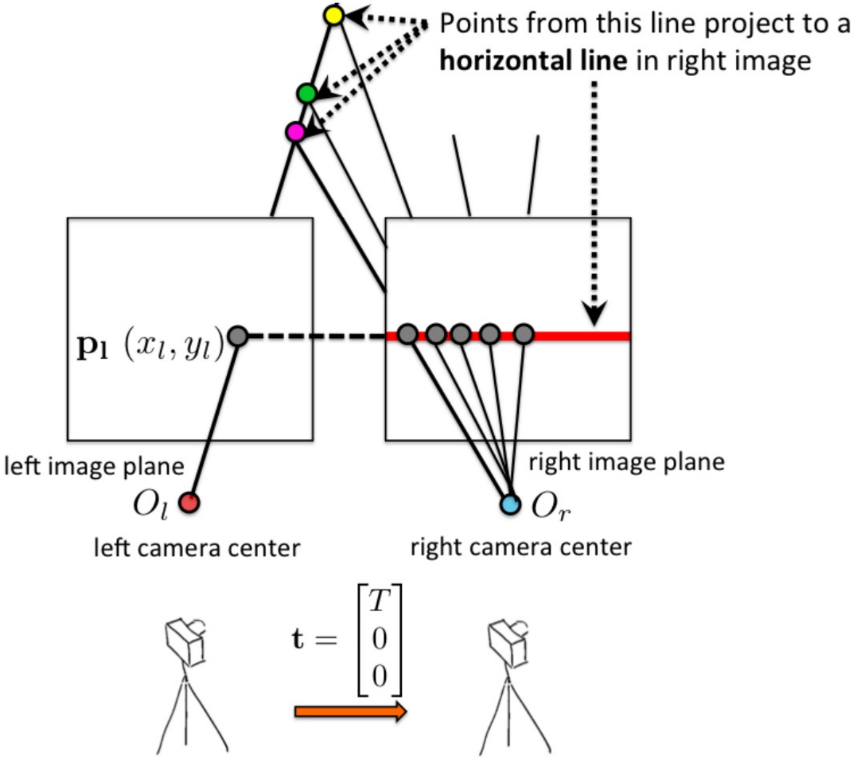
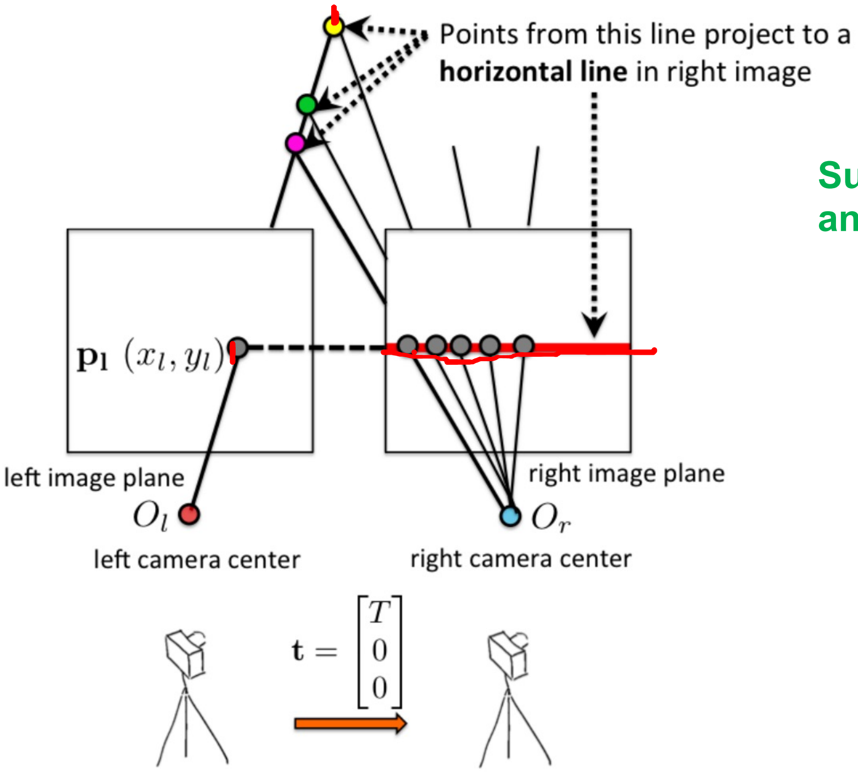


Image credit: Sanja Fidler

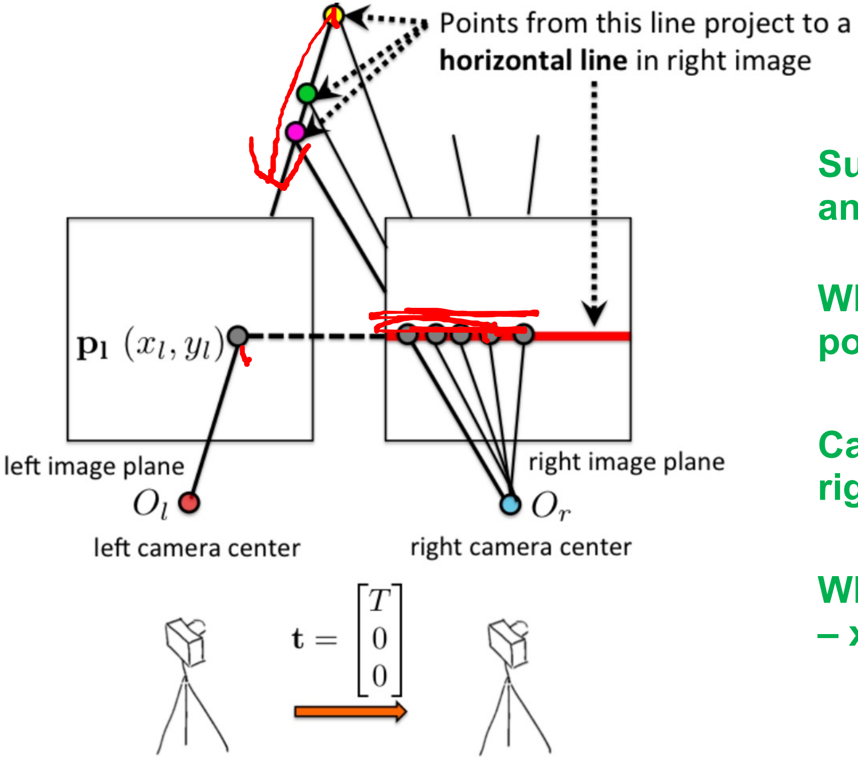
If Cameras are Frontal Parallel..



Suppose pl = (xl, yl) and pr = (xr, yr), could you tell anything about yr?

Image credit: Sanja Fidler

If Cameras are Frontal Parallel..



Suppose pl = (xl, yl) and pr = (xr, yr), could you tell anything about yr?

What is the right pixel location pr, if the 3D point is infinitely far from the two cameras?

Can the projected point on right camera fall right of p_l? (i.e. xr > xl?)

What is the relationship between the offset (xl – xr) and the distance to camera?

Image credit: Sanja Fidler

Binocular Vision

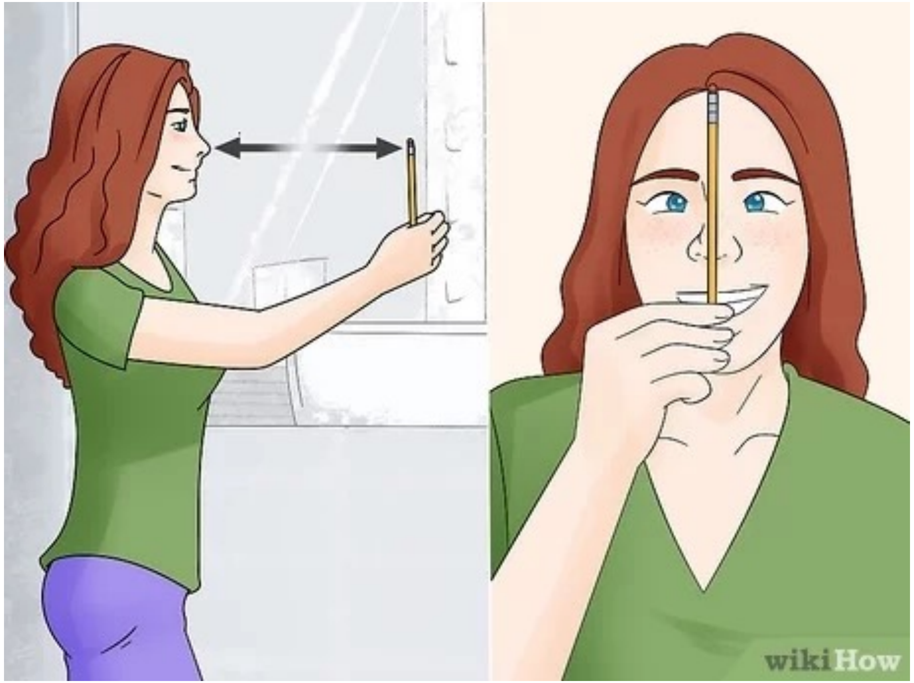
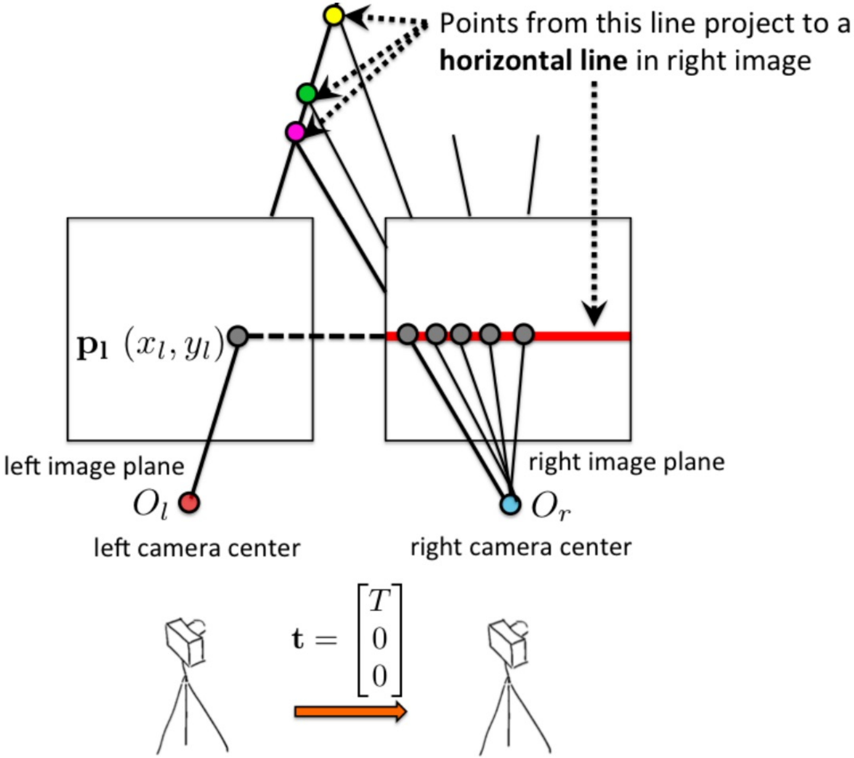


Image credit: wikiHow

Disparity vs Depth



What should I do if I want to see further?

focal length

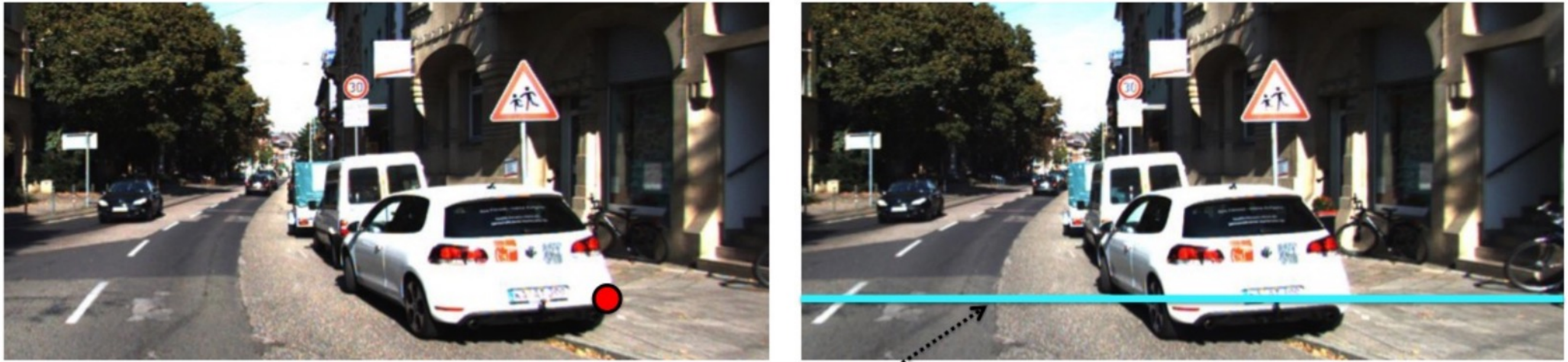
baseline

$$z = \frac{f \cdot T}{x_l - x_r}$$

depth

disparity

Stereo Estimation



left image

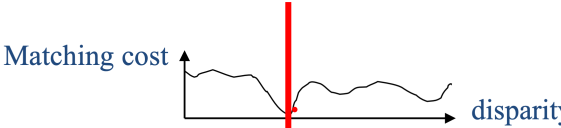
right image

the match will be on this line (same y)

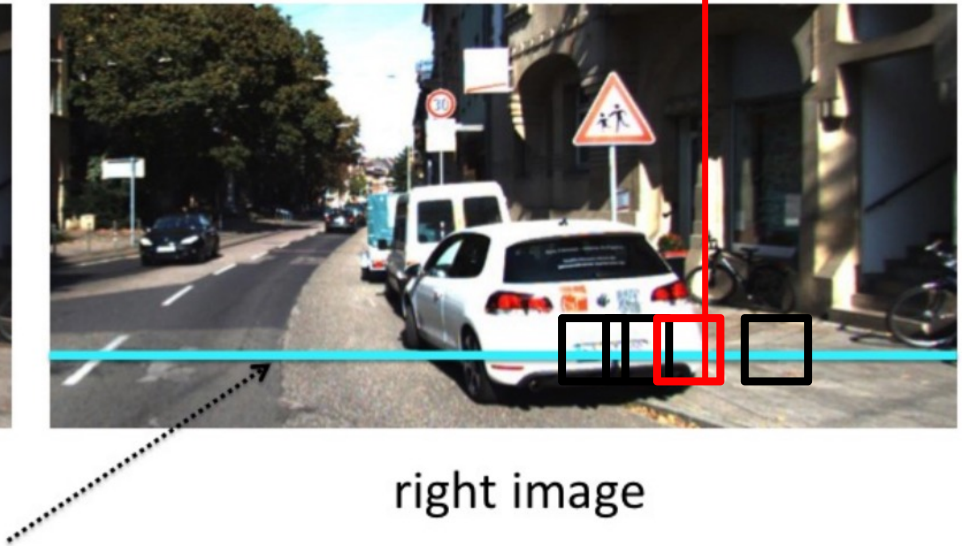
Image credit: KITTI

What would be a good matching function?

Correspondence Gives Disparity



left image

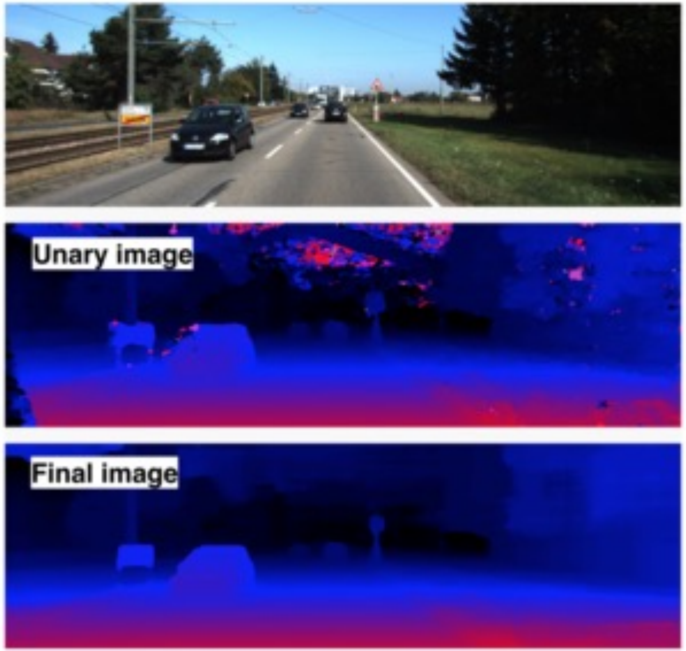


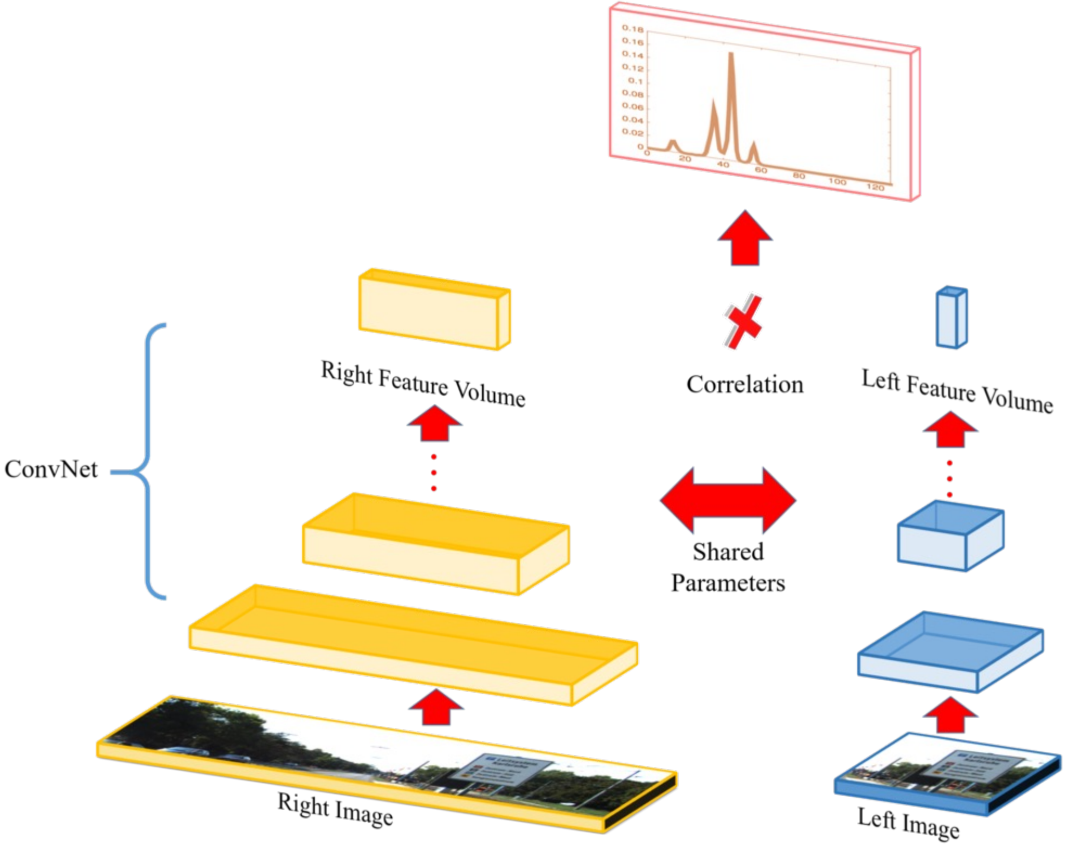
right image

Popular Methods: SSD, PatchMatch Stereo, SGM, etc.

Tutorial: https://docs.opencv.org/master/dd/d53/tutorial_py_depthmap.html

Stereo Estimation

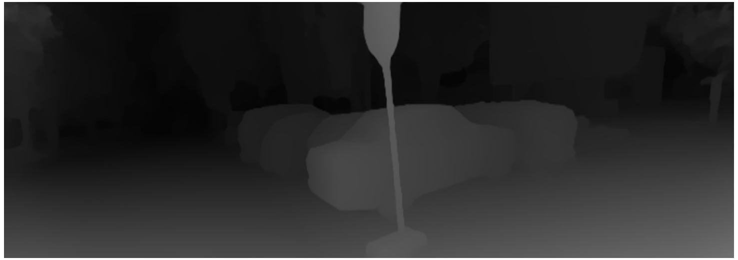




Luo et al. CVPR 16

Stereo Estimation

62ms



Duggal et al. DeepPruner, ICCV 19, Image produced by Yuan Shen

RGB Stereo Cameras

Pros:

- Works for both indoor and outdoor
- High-resolution

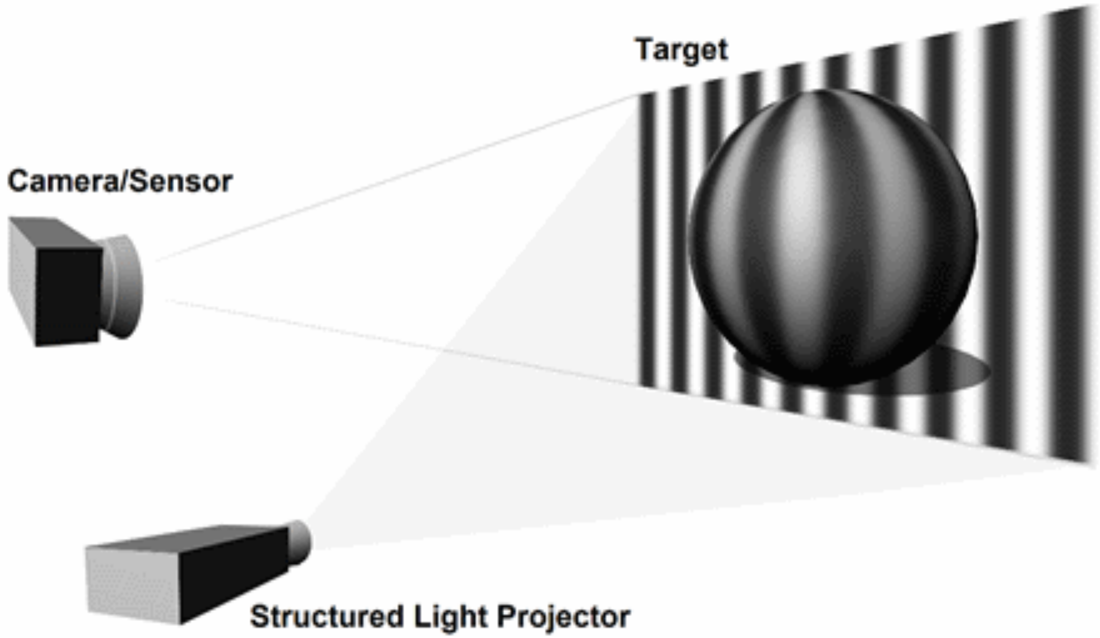
Cons:

- Less accurate for textureless region
- Not robust to specular reflected surface
- Cannot work in dark environment

Stereolabs Zed2

PointGrey Bubblebee

Structured Light Camera



Intel Realsense

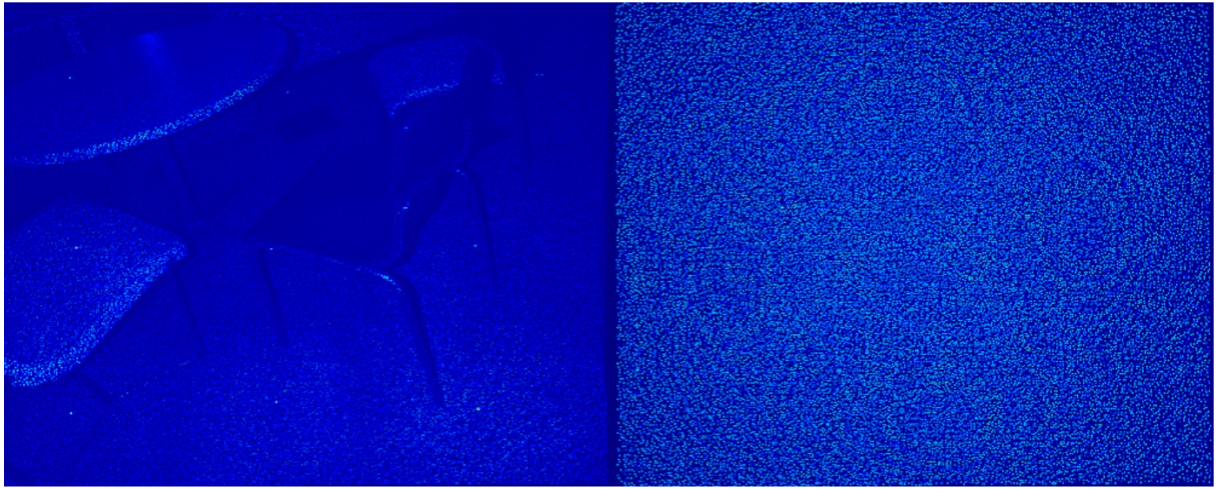
Microsoft Kinect

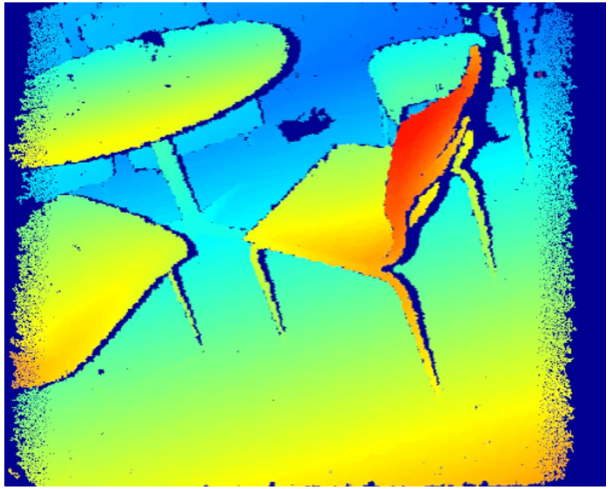
Structured Light Camera

Left: perceived near-infrared image

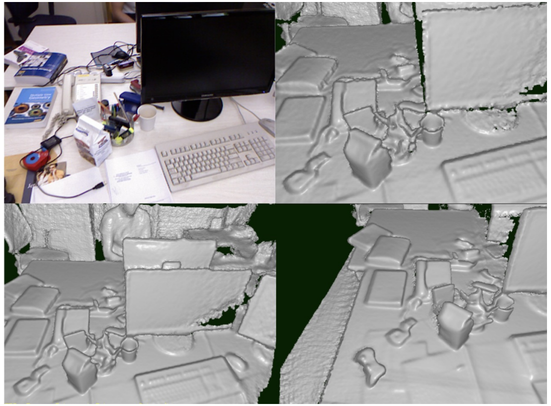
Right: structured light projector pattern

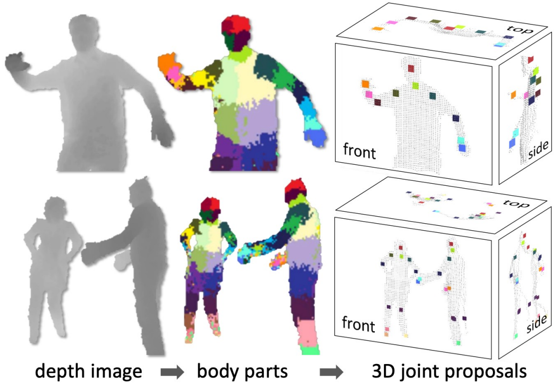
Depth

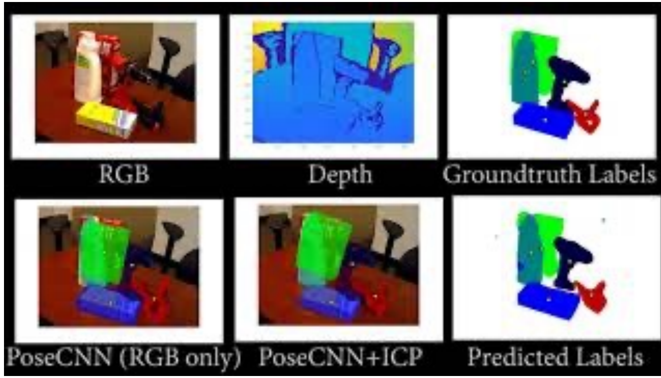




Structured Light Camera







KinectFusion

Articulated Shape Estimation

6DoF Pose Estimation

Structured Light Camera

Pros:

- Works in low light conditions
- Does not rely on having textured objects
- Not confused by repeated scene textures
- Can tailor algorithm to produced pattern

Cons:

- Does not work outside (influenced by sunlight)
- Interference to each other