NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

EE6222 Machine Vision

Topics 13/14
Vision Beyond Image 2:
3D Machine Vision (1)

Dr. Xu Yuecong

8 Nov 2023

Topics for Week 12 and 13

Topics Outline:

- Fundamentals of camera models. (13)
- Fundamentals of single-view geometry and stereo/epipolar geometry. (13)
- (Deep) Learning-based 3D vision analysis with multi-modal information. (14)
- Latest developments in 3D vision analysis (mostly with deep learning) (14)

Reference (Textbook):

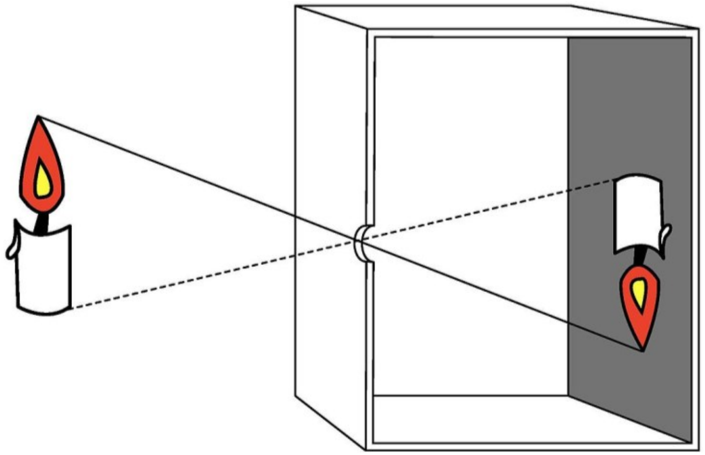
- Hartley, & Zisserman, A. (2004). Multiple View Geometry in Computer Vision. Cambridge University Press. https://doi.org/10.1017/CBO9780511811685 (Available in NTU Library)

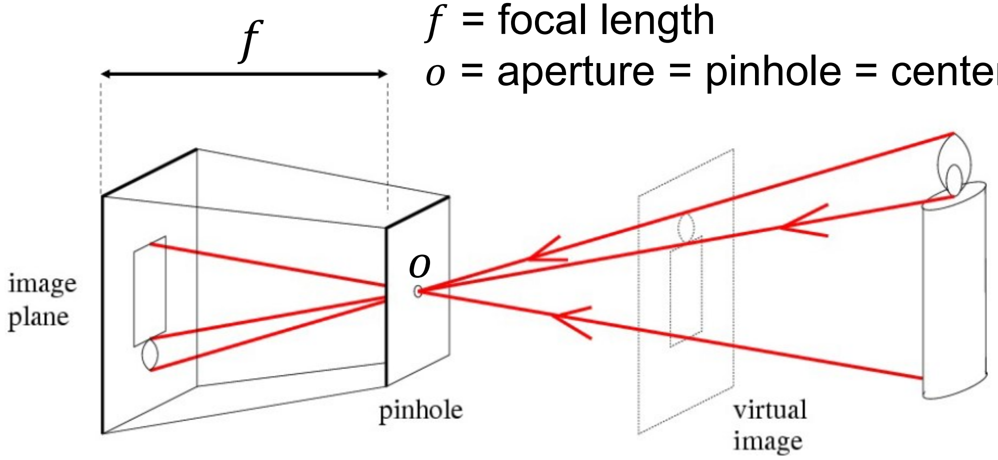
Goal of Topics 13 – 14

- ✓ Basic concepts and understanding in camera model, projection geometry, single-view/dual-view geometry. (Have been simplified to provide ONLY fundamental concepts and eqns.)
- ✓ Basic knowledge in the development of 3D analysis.
- ✓ Develop critical thinking in 3D analysis (*with vision and beyond*).

- ✓ **Understand**, not recitation.
- ✓ **Basic computation required**, understand how certain equations are formulated.
- ✓ Basic mathematical concepts needed (e.g., matrix multiplication, Taylor series approximation).

Camera Fundamentals: Projecting to Image Planes





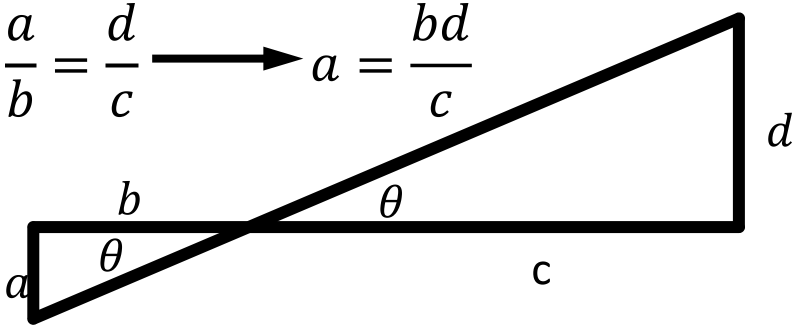
f = focal length
o = aperture = pinhole = center of the camera

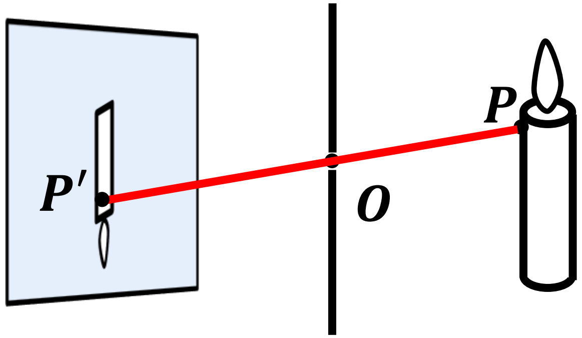
- The most fundamental type of cameras: pinhole camera.
  - ➢ Images formed are real and inverted – projection geometry built on pinhole camera model.
  - ➢ In reality, most cameras include a lens to focus the image on the plane.
  - ➢ Analyzing pinhole camera (and projection geometry) through similar triangles.

Right figure from CS231A Stanford University, Credit: Prof. Silvio Savarese

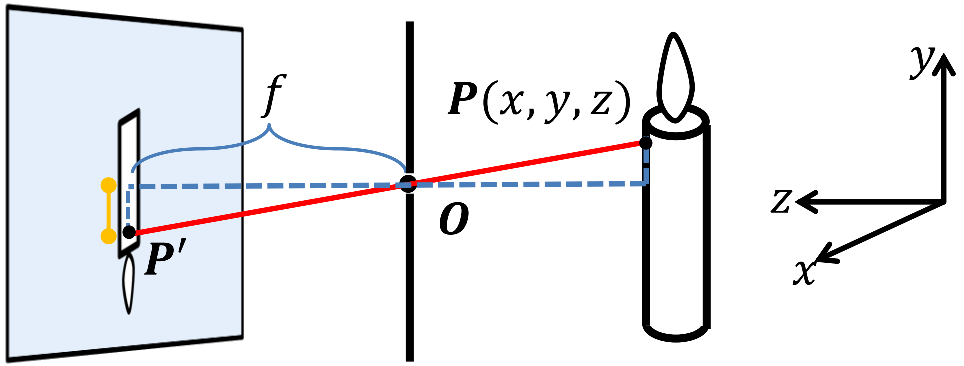
Camera Models: Pinhole Camera Analysis – Projection

$$\frac{a}{b} = \frac{d}{c} \longrightarrow a = \frac{bd}{c}$$





- The image formed through the pinhole camera is obtained by projecting each point in the 3D-space to the 2D-image plane.
- How to find the corresponding projected point for any point in the 3D-space?

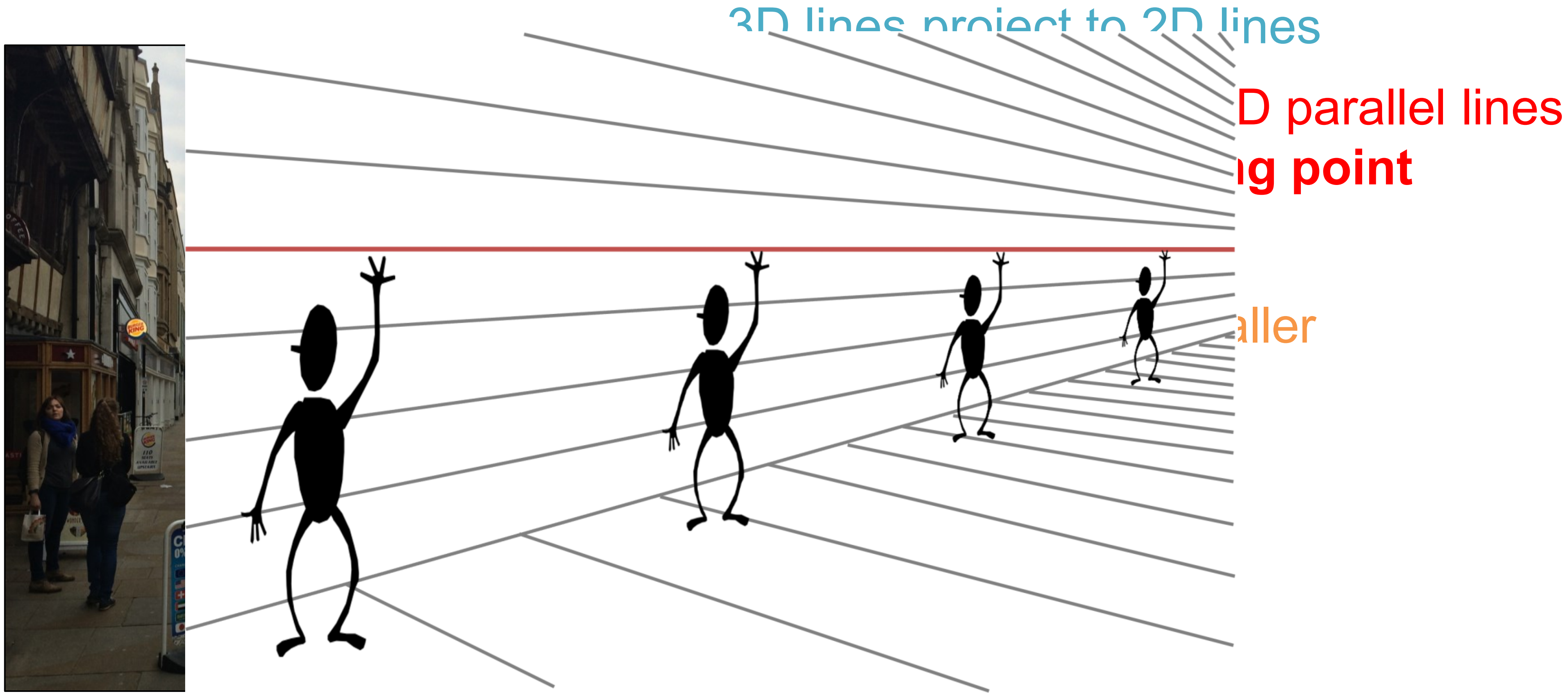


- Given point in 3D-space $\boldsymbol{P}(x, y, z)$, the corresponding point on the image plane is computed as:

$$\boldsymbol{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \boldsymbol{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} f\dfrac{x}{z} \\ f\dfrac{y}{z} \end{bmatrix}$$

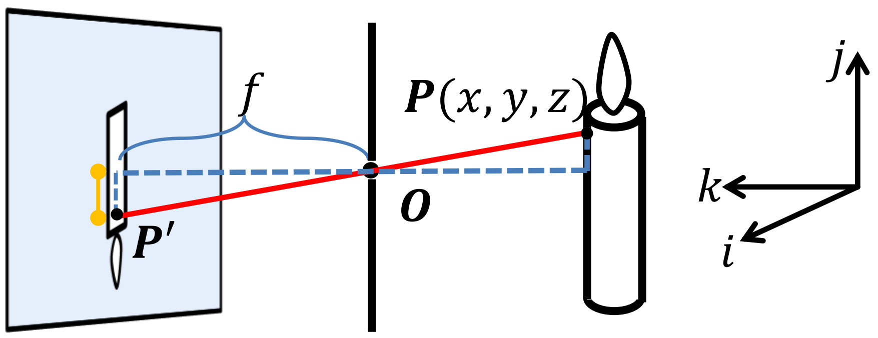
Figures from EECS 442 University of Michigan, Credit: Prof. Justin Johnson

Camera Models: Intuitive Concepts of Projection



Figures from EECS 442 University of Michigan, Credit: Prof. Justin Johnson

Camera Models: Projection Equation

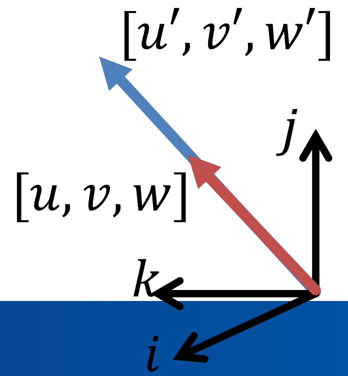


- Given point in 3D-space $\boldsymbol{P}(x, y, z)$, the point on the image plane is:

$$\boldsymbol{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \boldsymbol{P}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} f\frac{x}{z} \\ f\frac{y}{z} \end{bmatrix}$$

- Is this transformation linear? No! Division by z is Non-Linear – risks division by 0.
  - Division by 0 would result in points at infinity (related to vanishing points, more on that later).

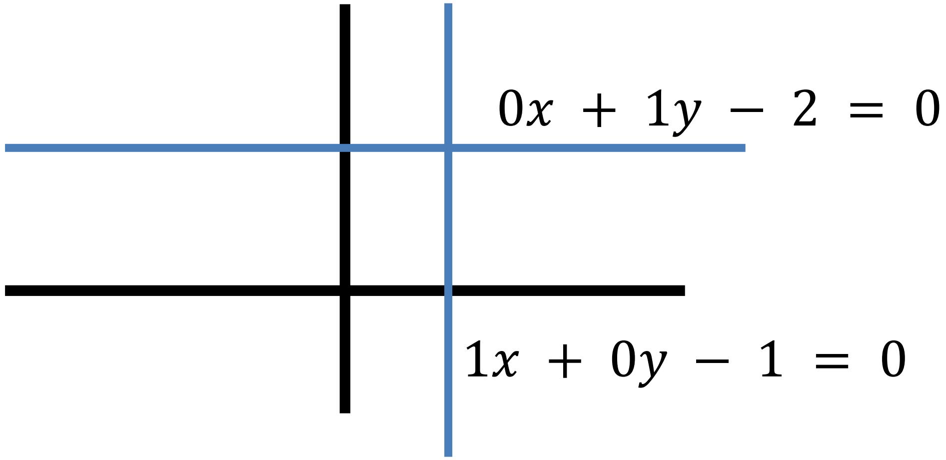
- How to mitigate the risk of zero-division?
- Homogeneous Coordinates (for 2D points first) – add a dimension:
- Physical → Homogeneous point → Physical Point: (Euclidean → Homogeneous → Euclidean)
- $\begin{bmatrix} x \\ y \end{bmatrix} \rightarrow \begin{bmatrix} u \\ v \\ w \end{bmatrix} \rightarrow \begin{bmatrix} \frac{u}{w} \\ \frac{v}{w} \end{bmatrix}$: usually we assume $w = 1$.
- Two homogeneous coordinates are equivalent (not equal) if they are proportional to each other – physically on the same line.
- $\begin{bmatrix} u \\ v \\ w \end{bmatrix} \equiv \begin{bmatrix} u' \\ v' \\ w' \end{bmatrix}$



Camera Models: Homogeneous Coordinates

- Why homogeneous coordinates?
- Lines (3D) and points (2D → 3D) are now the same dimension.
- Use the cross (×) and dot product for:
  - Intersection of lines l and m: $l \times m$
  - Line through two points p and q: $p \times q$
  - Point p on line l: $l^T p$.
- Express the general equation of a 2D line: $ax + by + c = 0$ with homogeneous coords:
- $l^T p = 0, l = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, p = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

- Example: Intersection of two lines

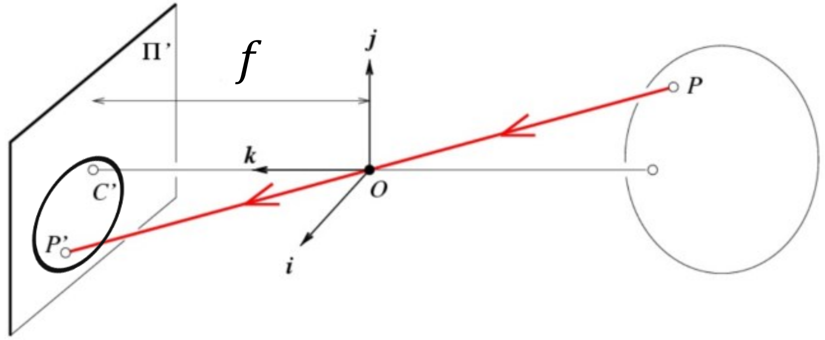


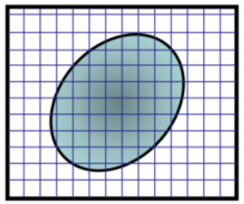
- $[0,1,-2] \times [1,0,-1] = [-1,-2,-1]$
- Convert back to physical point by dividing $w = -1$, the answer would be $[1,2]$

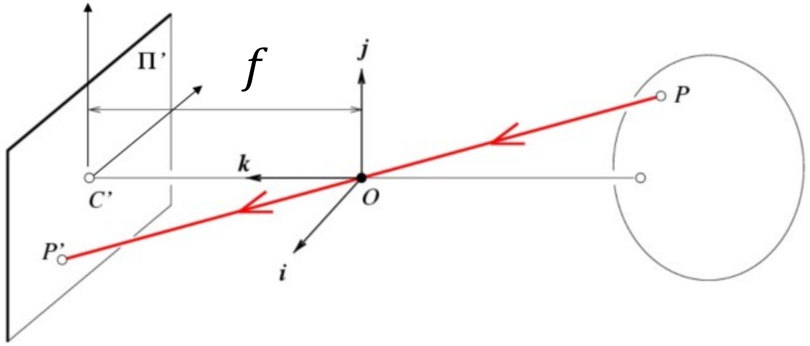
$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{vmatrix} = \begin{vmatrix} a_y & a_z \\ b_y & b_z \end{vmatrix} \mathbf{i} - \begin{vmatrix} a_x & a_z \\ b_x & b_z \end{vmatrix} \mathbf{j} + \begin{vmatrix} a_x & a_y \\ b_x & b_y \end{vmatrix} \mathbf{k}$$
$$= (a_y b_z - a_z b_y)\mathbf{i} + (a_z b_x - a_x b_z)\mathbf{j} + (a_x b_y - a_y b_x)\mathbf{k}$$

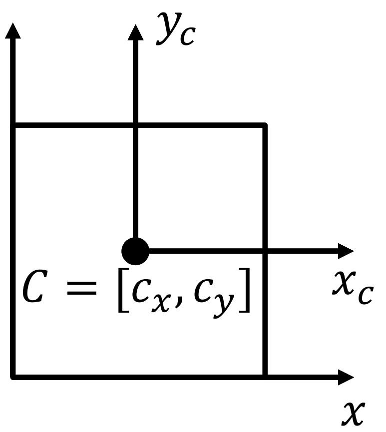
Right figure from EECS 442 University of Michigan, Credit: Prof. Justin Johnson

Camera Models: Projection Matrix





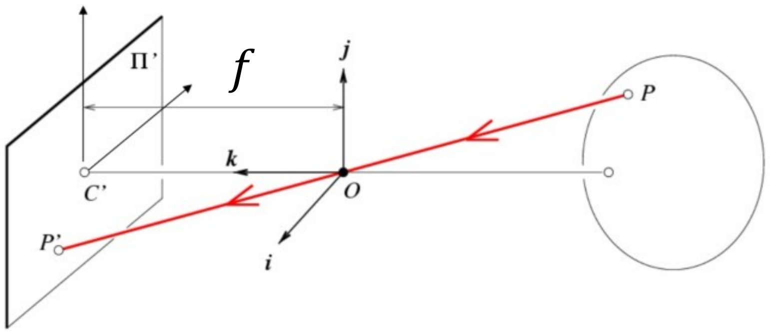


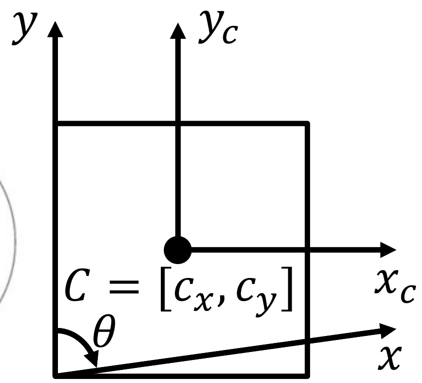


- $\boldsymbol{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \to \boldsymbol{P'} = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} f\frac{x}{z} \\ f\frac{y}{z} \end{bmatrix}$; $\mathbb{R}^3 \to \mathbb{R}^2$
- This is the most ideal condition – in reality, there will be a set of factors that effects where and how to the actual 3D world point is projected on the **digital** 2D image/camera plane.
- Digital image: discrete pixels, default bottom-left coordinate systems
- Factor 1: offset of image plane center in the image plane coordinate.
- $\boldsymbol{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \to \boldsymbol{P'} = \begin{bmatrix} f\frac{x}{z} + c_x \\ f\frac{y}{z} + c_y \end{bmatrix}$
- Factor 2: changing from metric to pixels (f – meters, computation in image planes are pixels (non-square)).
- $\boldsymbol{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \to \boldsymbol{P'} = \begin{bmatrix} \overbrace{fk}^{\alpha}\frac{x}{z} + c_x \\ \underbrace{fl}_{\beta}\frac{y}{z} + c_y \end{bmatrix} = \begin{bmatrix} \alpha\frac{x}{z} + c_x \\ \beta\frac{y}{z} + c_y \end{bmatrix} \to \begin{bmatrix} \alpha x + c_x z \\ \beta y + c_y z \\ z \end{bmatrix}$

Camera Models: Projection Matrix

- $\boldsymbol{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \boldsymbol{P}' = \begin{bmatrix} fk\frac{x}{z} + c_x \\ fl\frac{y}{z} + c_y \end{bmatrix} = \begin{bmatrix} \alpha\frac{x}{z} + c_x \\ \beta\frac{y}{z} + c_y \end{bmatrix} \rightarrow \begin{bmatrix} \alpha x + c_x z \\ \beta y + c_y z \\ z \end{bmatrix}$
- We express the projection from the 3D world space to the 2D camera/image plane with homogeneous coord:
- $\boldsymbol{P_h}' = \begin{bmatrix} \alpha x + c_x z \\ \beta y + c_y z \\ z \end{bmatrix} = \boldsymbol{MP_h} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \boldsymbol{K}[\boldsymbol{I} \quad 0]\boldsymbol{P}$
- $\boldsymbol{M} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$: Camera Model Matrix.
- $\boldsymbol{K} = \begin{bmatrix} \alpha & 0 & c_x \\ 0 & \beta & c_y \\ 0 & 0 & 1 \end{bmatrix}$: Camera Intrinsic Matrix – determined by camera intrinsics: focal points and optical center.

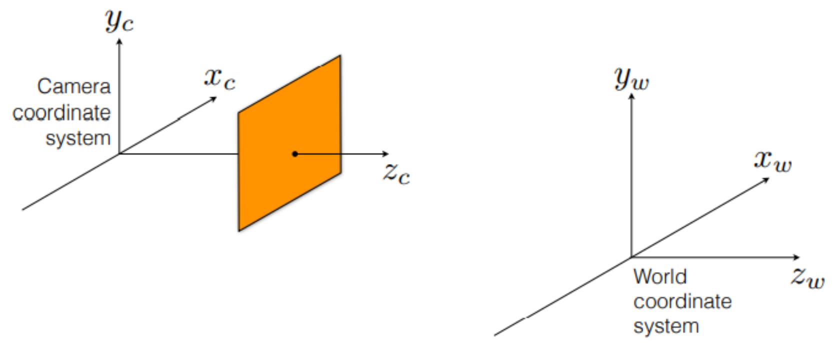




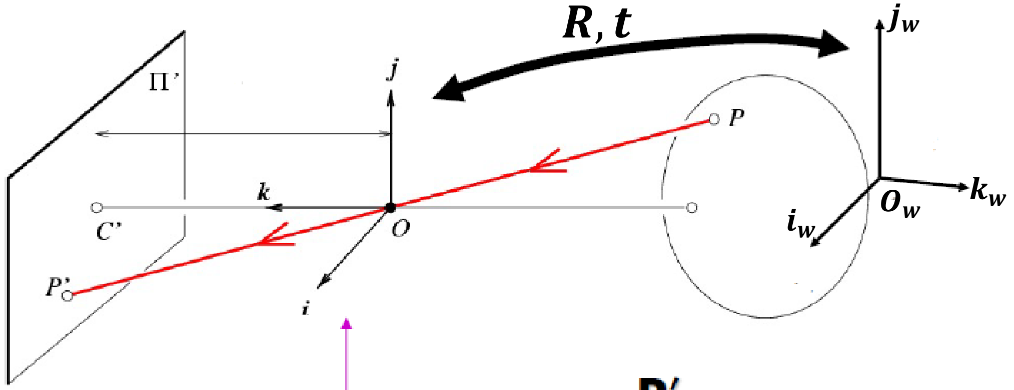
- Factor 3: camera skewness – not perfect aligned pixels.
- Omitting the subscript of homogeneous coord:
- $\boldsymbol{P}' = \boldsymbol{MP} = \begin{bmatrix} \alpha & -\alpha\cot\theta & c_x & 0 \\ 0 & \beta/\sin\theta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

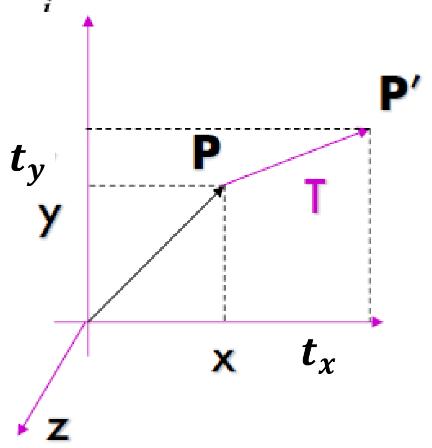
Right figure from CS231A Stanford University, Credit: Prof. Silvio Savarese

Camera Models: Projection Transformation with Differed World Coordinate Systems



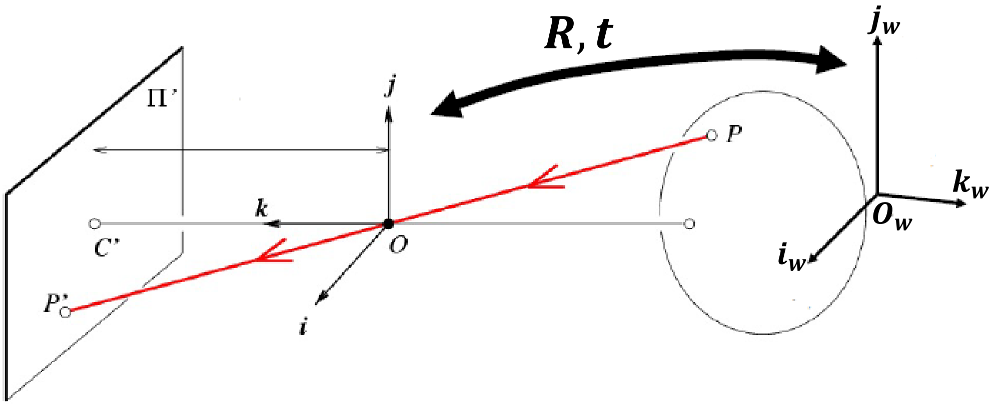
- The above matrices assume: the camera and world share the same coordinate system.
- What if they are different? How do we align a 3D point in the world coordinate system with a 2D point in the camera coordinate system?
- Align the two coordinates through 3D rotation and translation!
- 1. 3D Translation
- Define the 3D Translation matrix as: $\boldsymbol{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$
- $\boldsymbol{P'} \rightarrow \boldsymbol{P} + \boldsymbol{t} = (x + t_x, y + t_y, z + t_z)$ (→: corresponds to)
- $\boldsymbol{P'} \rightarrow \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$





Top figure from 16-385 Carnegie Mellon University, Credit: Prof. Kris Kitani
Bottom figure from CS231A Stanford University, Credit: Prof. Silvio Savarese

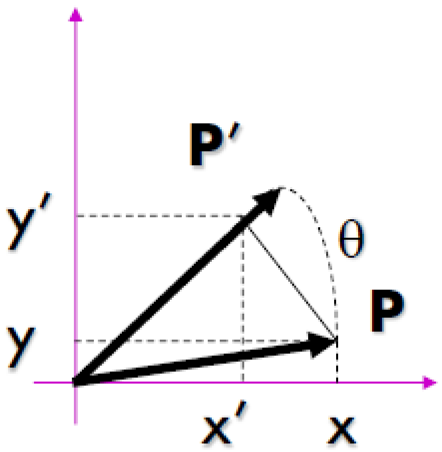
Camera Models: Projection Transformation with Differed World Coordinate Systems

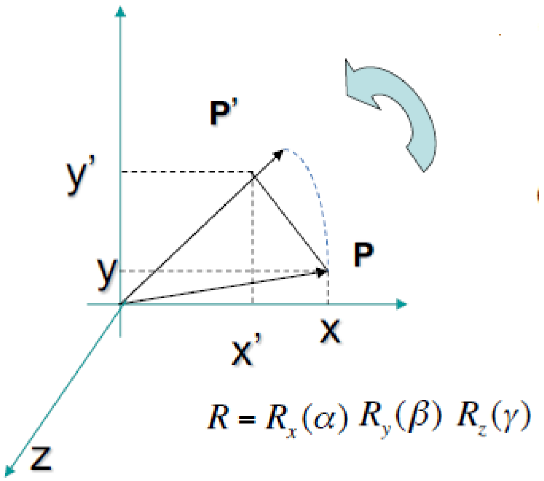


- 2. 3D Rotation
- A rotation matrix $\boldsymbol{R}_x(\alpha)$ corresponds to the rotation around the x-axis anti-clockwise, over α degrees
- Start from a 2D rotation first (via trigonometric functions):
- $\begin{cases} x' = \cos\theta x - \sin\theta y \\ y' = \cos\theta x + \sin\theta y \end{cases} \rightarrow \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$
- Similarly, we obtain the rotation matrix corresponding to rotating around the three axis as:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \quad R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \quad R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

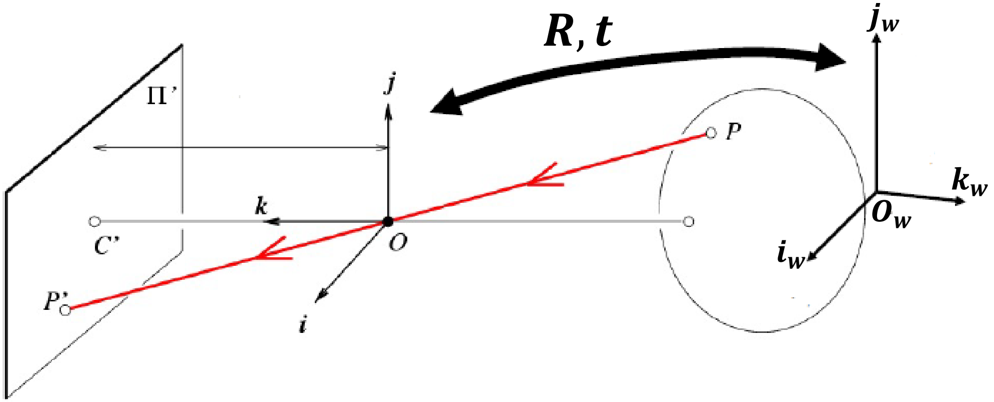
- The rotation matrix is defined as: $\boldsymbol{R} = \boldsymbol{R}_x(\alpha)\boldsymbol{R}_y(\beta)\boldsymbol{R}_z(\gamma)$.
- $\boldsymbol{P}' \rightarrow \begin{bmatrix} \boldsymbol{R} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$





Camera Models: Projection Transformation with Differed World Coordinate Systems

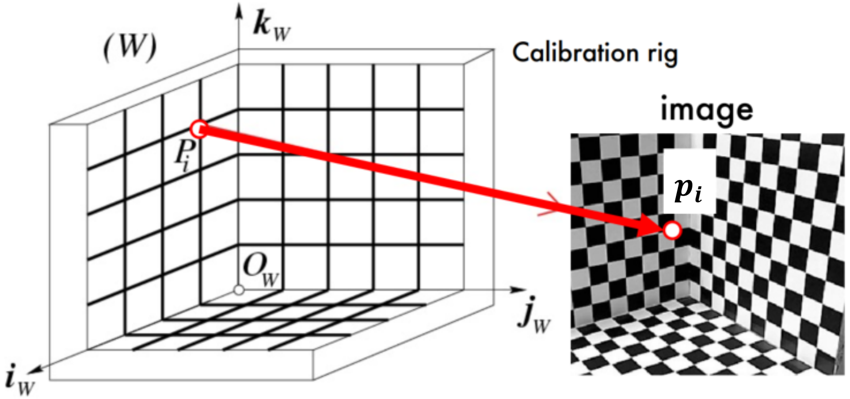
- Combining the translation matrix and rotation matrix:
- $$P' \rightarrow \begin{bmatrix} I & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$
- Denote $\boldsymbol{P_w}$ as the 3D point $\boldsymbol{P}$ in the world coordinate system, and $\boldsymbol{P}$ is defined in the camera coordinate system, we have:
- $$\boldsymbol{P} = \begin{bmatrix} I & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{P_w} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \boldsymbol{P_w}$$
- From previous slides, we have $\boldsymbol{P}' = \boldsymbol{MP} = \boldsymbol{K}[\boldsymbol{I} \quad 0]\boldsymbol{P}$. Sub in $\boldsymbol{P_w}$ we can therefore transform a 3D point measured in the world coordinate system to the 2D image point in the camera coordinate system by:
- $$\boldsymbol{P}' = \boldsymbol{K}[\boldsymbol{I} \quad 0]\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{I} \quad 0] \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \boldsymbol{P_w} = \boldsymbol{K}[\boldsymbol{R} \quad \boldsymbol{t}]\boldsymbol{P_w} = \boldsymbol{K}[\boldsymbol{R}, \boldsymbol{t}]\boldsymbol{P_w} = \boldsymbol{MP_w}$$
- $\boldsymbol{P}' \rightarrow \begin{bmatrix} A & t \\ v & b \end{bmatrix} \boldsymbol{P_w}$ (homogeneous coordinate), where $\begin{bmatrix} A & t \\ v & b \end{bmatrix}$ is the general form of a projection matrix H.



- $\boldsymbol{P}' = \boldsymbol{K}[\boldsymbol{R}, \boldsymbol{t}]\boldsymbol{P_w}$
- In some literature refer to as the perspective model.
- Define:
  - $\boldsymbol{K}$ – Camera Intrinsic Matrix;
  - $[\boldsymbol{R}, \boldsymbol{t}]$ – Extrinsic Matrix (default refer to the extrinsic matrix of the world coordinate system with respect to the camera coordinate system.

Camera Models: Camera Calibration

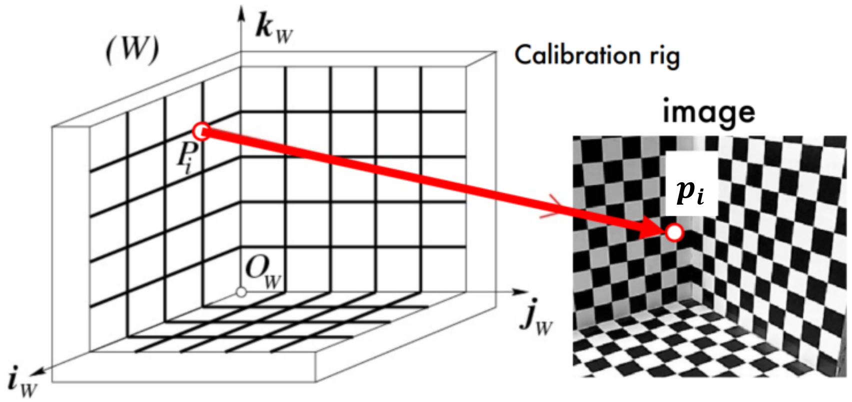
- Camera calibration is essentially finding the camera model matrix given the 3D point in the world coordinate system and the 2D point on the image plane in the camera coordinate system.
- $\boldsymbol{P}' = \boldsymbol{K}[\boldsymbol{R}, \boldsymbol{t}]\boldsymbol{P_w} = \boldsymbol{M}\boldsymbol{P_w}$; $\boldsymbol{M}$ – camera model matrix with both intrinsic and extrinsic components (size 3×4).
- We can re-write the above equation into: (by converting from homogeneous coord. to Euclidean coord.)
- $$\boldsymbol{P}' = \boldsymbol{M}\boldsymbol{P_w} = \begin{bmatrix} \boldsymbol{m_1} \\ \boldsymbol{m_2} \\ \boldsymbol{m_3} \end{bmatrix} \boldsymbol{P_w} = \begin{bmatrix} \boldsymbol{m_1 P_w} \\ \boldsymbol{m_2 P_w} \\ \boldsymbol{m_3 P_w} \end{bmatrix} \rightarrow \begin{bmatrix} \frac{\boldsymbol{m_1 P_w}}{\boldsymbol{m_3 P_w}} \\ \frac{\boldsymbol{m_2 P_w}}{\boldsymbol{m_3 P_w}} \end{bmatrix}$$



- If we denote a point in space as point P_i and its corresponding point in the image as point p_i, we then have:
- $$p_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} \frac{\boldsymbol{m_1} P_i}{\boldsymbol{m_3} P_i} \\ \frac{\boldsymbol{m_2} P_i}{\boldsymbol{m_3} P_i} \end{bmatrix} = \boldsymbol{M} P_i; \text{ equivalently } \begin{cases} u_i = \frac{\boldsymbol{m_1} P_i}{\boldsymbol{m_3} P_i} \ or\ u_i(\boldsymbol{m_3} P_i) - \boldsymbol{m_1} P_i = 0 \\ v_i = \frac{\boldsymbol{m_2} P_i}{\boldsymbol{m_3} P_i} \ or\ v_i(\boldsymbol{m_3} P_i) - \boldsymbol{m_2} P_i = 0 \end{cases}$$

Camera Models: Camera Calibration

- $\boldsymbol{M} = \boldsymbol{K}[\boldsymbol{R}, \boldsymbol{t}]$; how many degrees of freedom does $\boldsymbol{M}$ have?
  - 5 (from $\boldsymbol{K}$) + 3 (from $\boldsymbol{R}$) + 3 (from $\boldsymbol{t}$) = 11
- One pair of corresponding 3D-2D points result in 2 equations – as long as you have n points with $2n > 11$, the homogeneous linear system is overdetermined – $\boldsymbol{M} = \boldsymbol{0}$ is always a trivial solution.
- No Math, just FYI!
- To constrain the solution, we obtain $\boldsymbol{M}$ with the following objectives:
  - Minimize $\sum ||\boldsymbol{M}P_i||^2$
  - Subject to $||\boldsymbol{M}||^2 = 1$
- The Intrinsic and Extrinsic Matrices are obtained from $\boldsymbol{M}$.

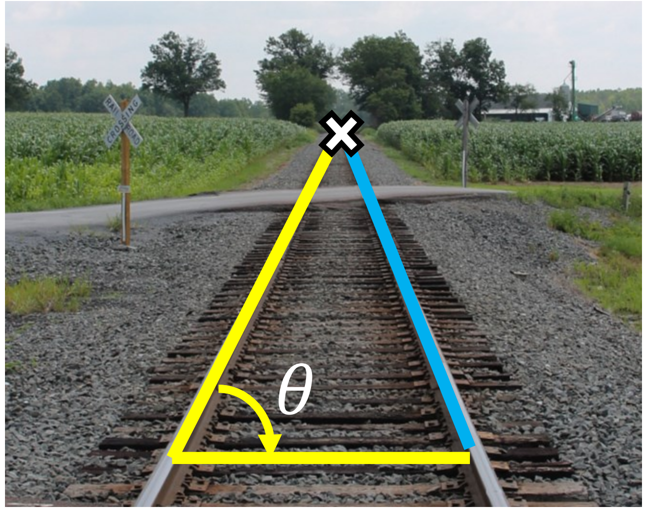


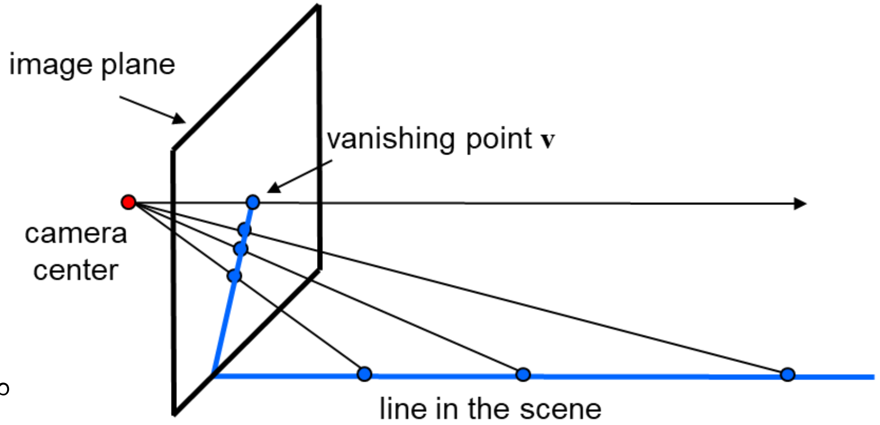
Single-view Geometry: Vanishing Points

- Properties of Projection:
  - Angles are NOT preserved.
  - Parallel lines meet – parallel lines in the world intersect in the image at a **"vanishing point"**. Vanishing point – projection of a "point at infinity".
- "Point at infinity" and vanishing point:
  - A point at infinity in the 3D world space can be written in the homogeneous form of $p_\infty = \begin{bmatrix} x_\infty \\ y_\infty \\ z_\infty \\ 0 \end{bmatrix}$, e.g., $\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$ is the point at infinity relevant to the 3D line $x + y + z = 0$.
  - If we apply the general form of the projection matrix to the point at infinity, the projected point is computed as:

$$p'_\infty = Hp_\infty = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} \\ v & b \end{bmatrix} \begin{bmatrix} x_\infty \\ y_\infty \\ z_\infty \\ 0 \end{bmatrix} = \begin{bmatrix} x'_\infty \\ y'_\infty \\ z'_\infty \\ q'_\infty \end{bmatrix}$$

  - p'_∞ corresponds to a point in the image plane by $p'_\infty \rightarrow \boldsymbol{K}[\boldsymbol{I} \ \ \boldsymbol{0}]p'_{i\infty}$





Single-view Geometry: Vanishing Points

- "Point at infinity" and vanishing point:
  - $$p'_\infty = Hp_\infty = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{t} \\ v & b \end{bmatrix} \begin{bmatrix} x_\infty \\ y_\infty \\ z_\infty \\ 0 \end{bmatrix} = \begin{bmatrix} x'_\infty \\ y'_\infty \\ z'_\infty \\ q'_\infty \end{bmatrix}$$
  - A generic projective transformation maps points at infinity to points that are no longer at infinity – their corresponding points on the image plane are the vanishing points.
  - If a 3D line is defined as $\boldsymbol{d} = [a, b, c]^T$, then the corresponding vanishing point is computed by:
  - $\mathbf{v} = \boldsymbol{Kd}$

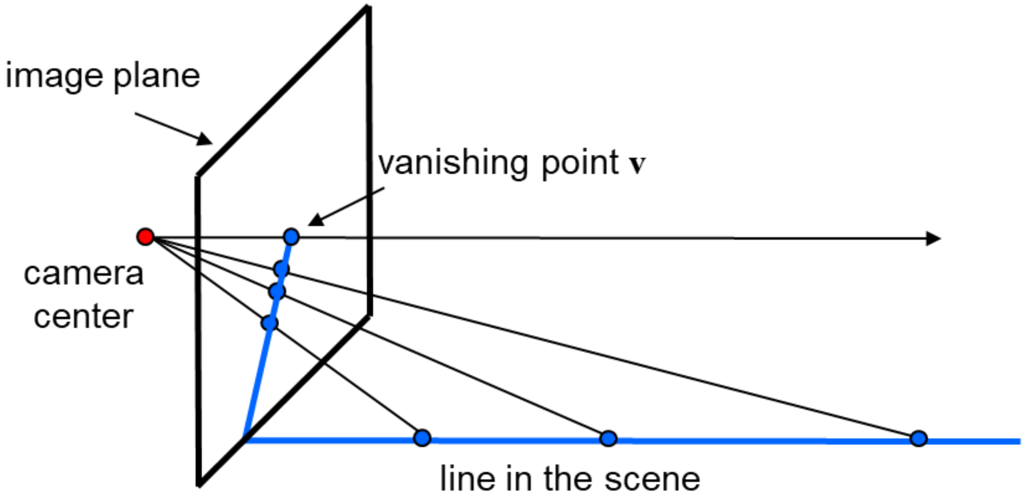


Figure from EECS 442 University of Michigan, Credit: Prof. Justin Johnson

Single-view Geometry: Vanishing Line/Horizon Line

- Vanishing Line/Horizon Line:
  - A line at infinity is a line where two parallel planes intersect.
  - The projective transformation of the line at infinity is no longer at infinity, but at the vanishing line, or the horizon line.
  - The horizon line is essentially a line that passes through the corresponding vanishing points in the image.
  - Often used as a starting line or reference line for 3D reconstruction/measurement.

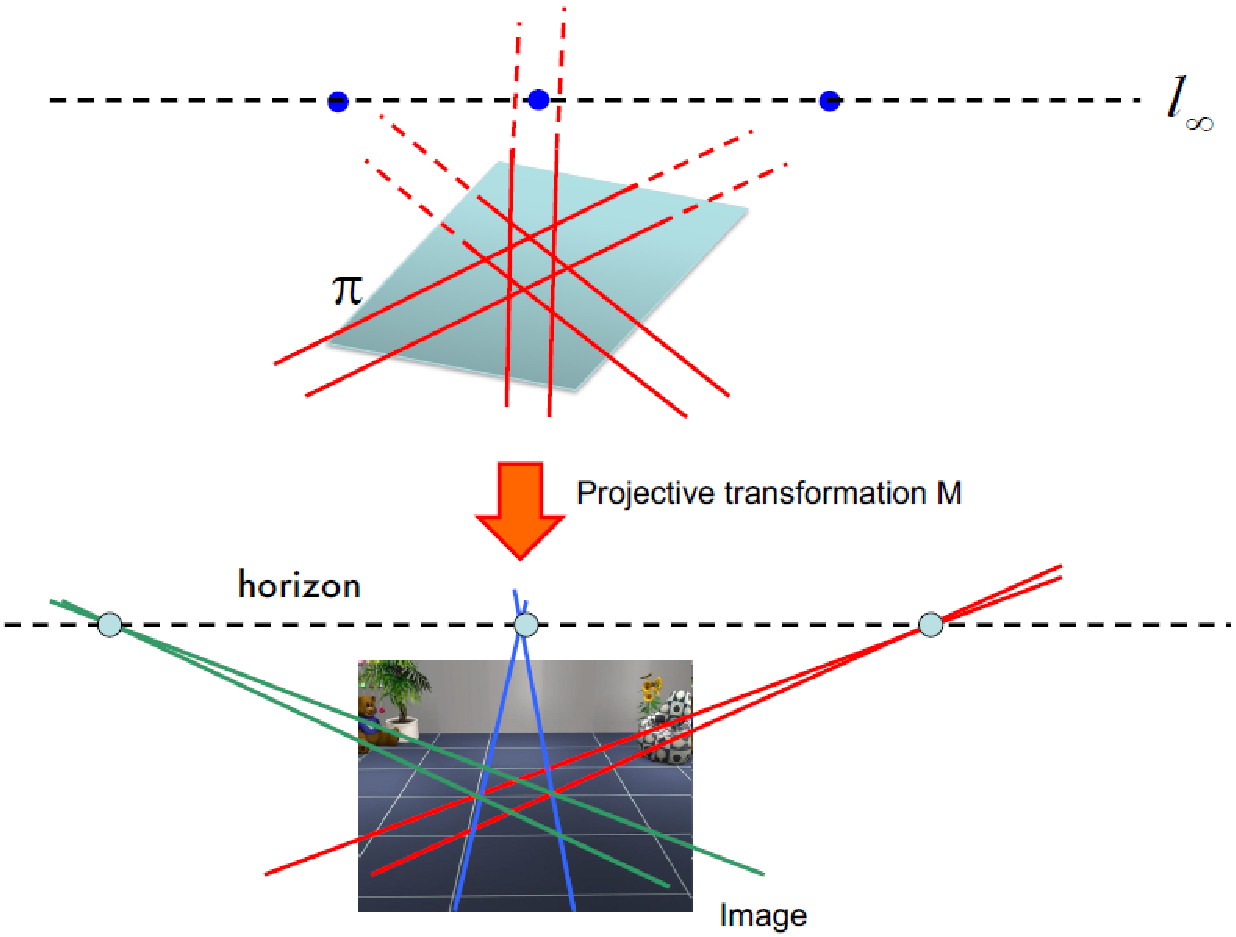


Figure from CS231A Stanford University, Credit: Prof. Silvio Savarese

Single-view Geometry: Measure with Vanishing Point/Line

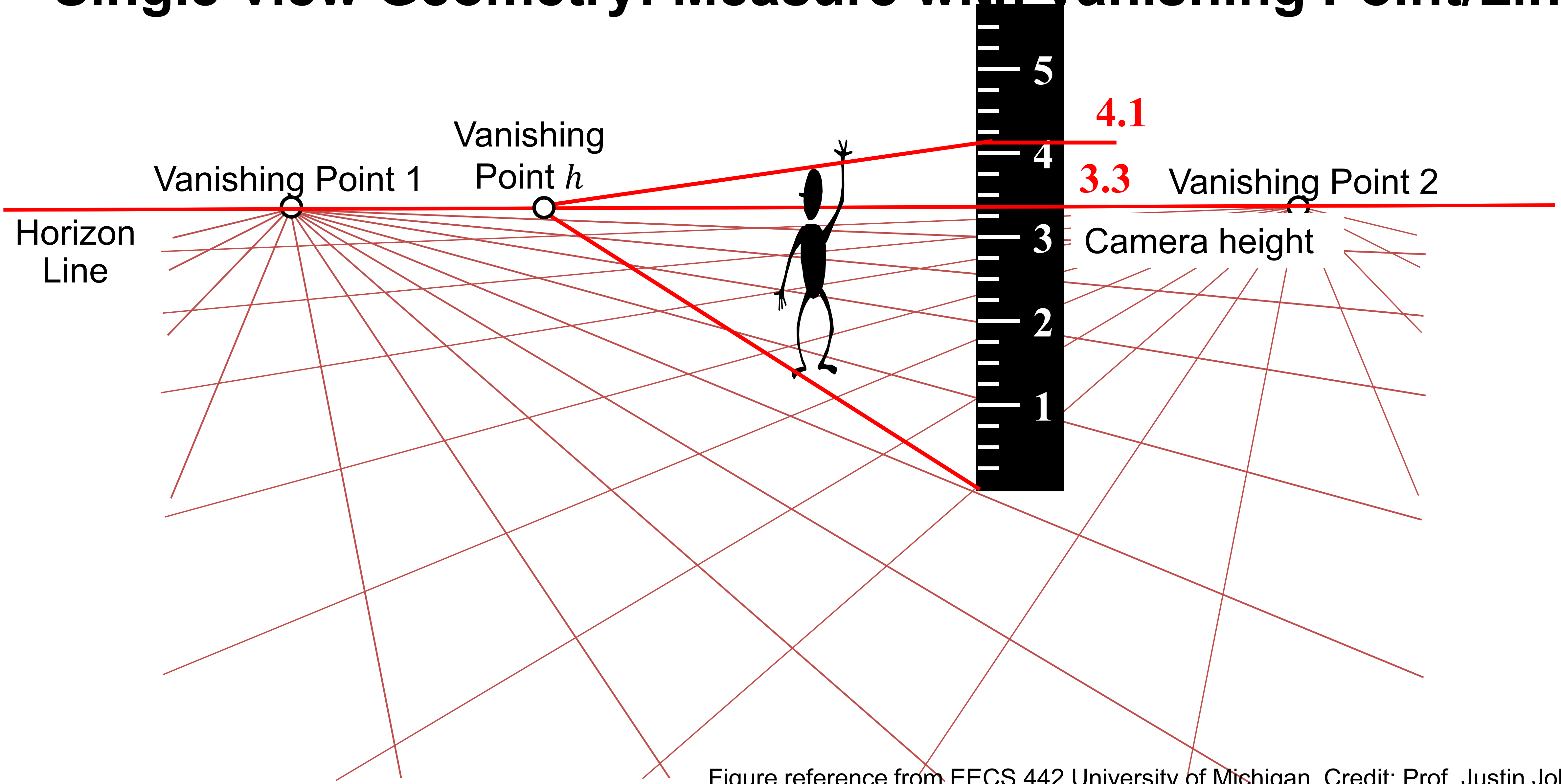


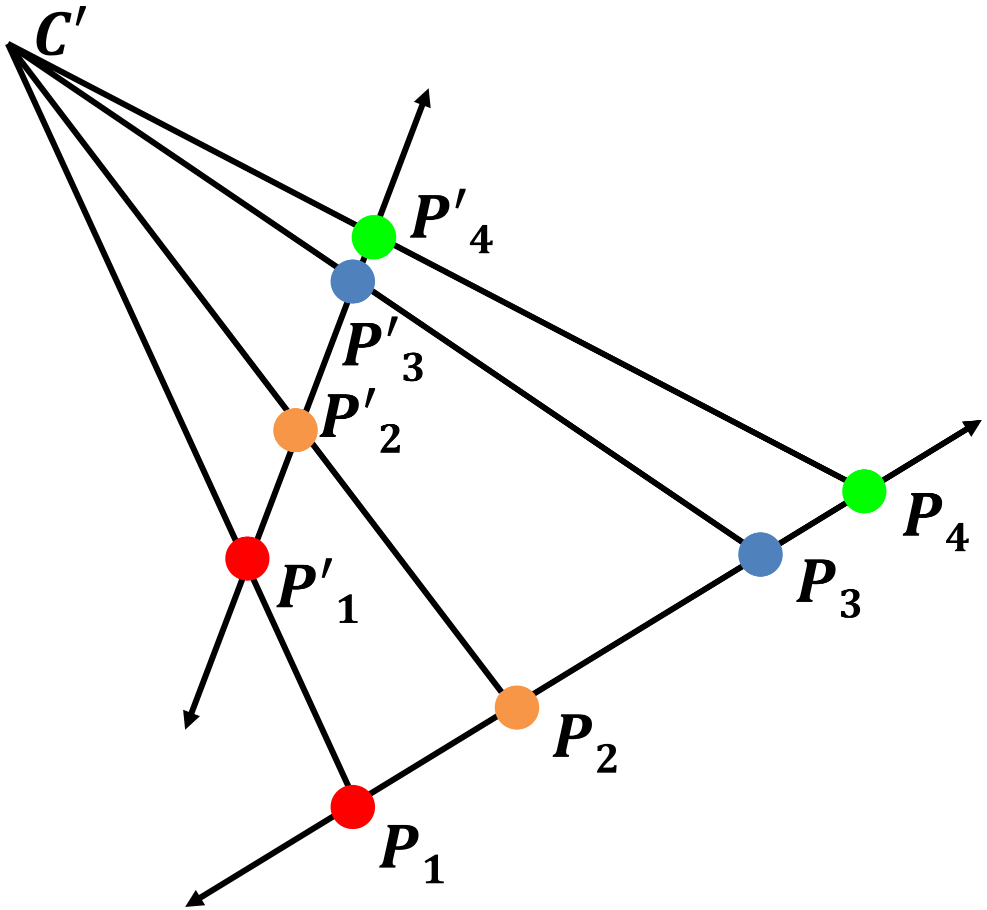
Figure reference from EECS 442 University of Michigan, Credit: Prof. Justin Johnson

Single-view Geometry: Measuring w/o Ruler – Projective Invariant

- Can we find the actual length of w in the world space from the given image **without** a ruler?
- Need more than vanishing points!
- *Projective invariant*: a quantity that does not change under projective transformations.

Single-view Geometry: Projective Invariant – Cross Ratio

- *Theorem: we denote the cross ratio of the four points in space as* $[P_1, P_2, P_3, P_4]$*, while the correspond to a set of projected points on the image space* P'_1, P'_2, P'_3, P'_4*. One of its cross ratio is defined as* $[P_1, P_2, P_3, P_4] = \frac{||P_1P_3||||P_2P_4||}{||P_2P_3||||P_1P_4||} = \frac{||P_1P_3||}{||P_2P_3||} / \frac{||P_1P_4||}{||P_2P_4||}$*. This cross ratio is projective invariant, that is:* $[P_1, P_2, P_3, P_4] = [P'_1, P'_2, P'_3, P'_4]$.



Single-view Geometry: Measuring w/o Ruler – Projective Invariant

- Since cross ratio is project invariant, we thus have $[A, B, C, D] = [A', B', C', D']$, that is:

$$\frac{||CB||\,||AD||}{||AB||\,||CD||} = \frac{||C'B'||\,||A'D'||}{||A'B'||\,||C'D'||}$$

- Therefore, the following equation stands:

$$\frac{(10+20)(20+30)}{20(10+20+30)} = \frac{(7+w)(w+6)}{w(7+w+6)}$$

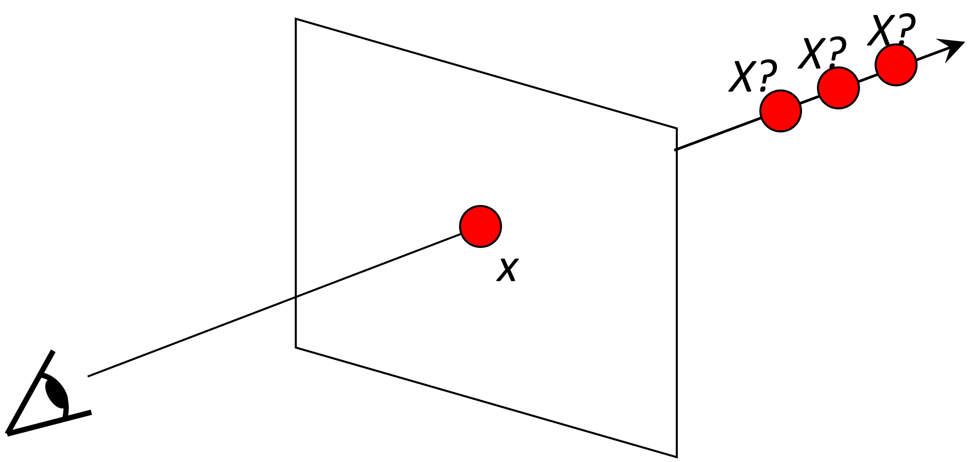
That is:

$$\frac{5}{4} = \frac{w^2+13w+42}{w^2+13w}$$

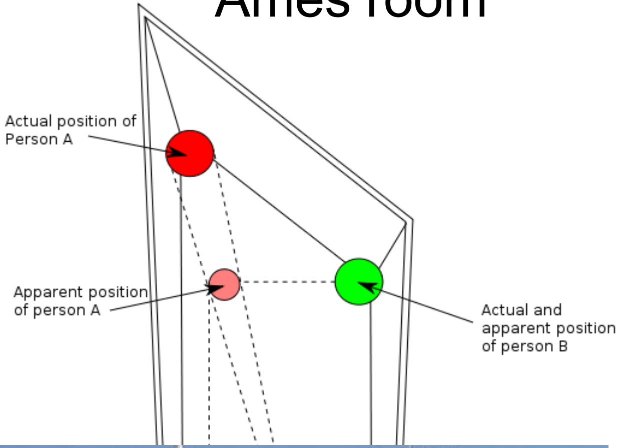
Therefore

$$w = 8\text{m}\ (w > 0)$$

Problem with Single-view Geometry: Single-view Ambiguity



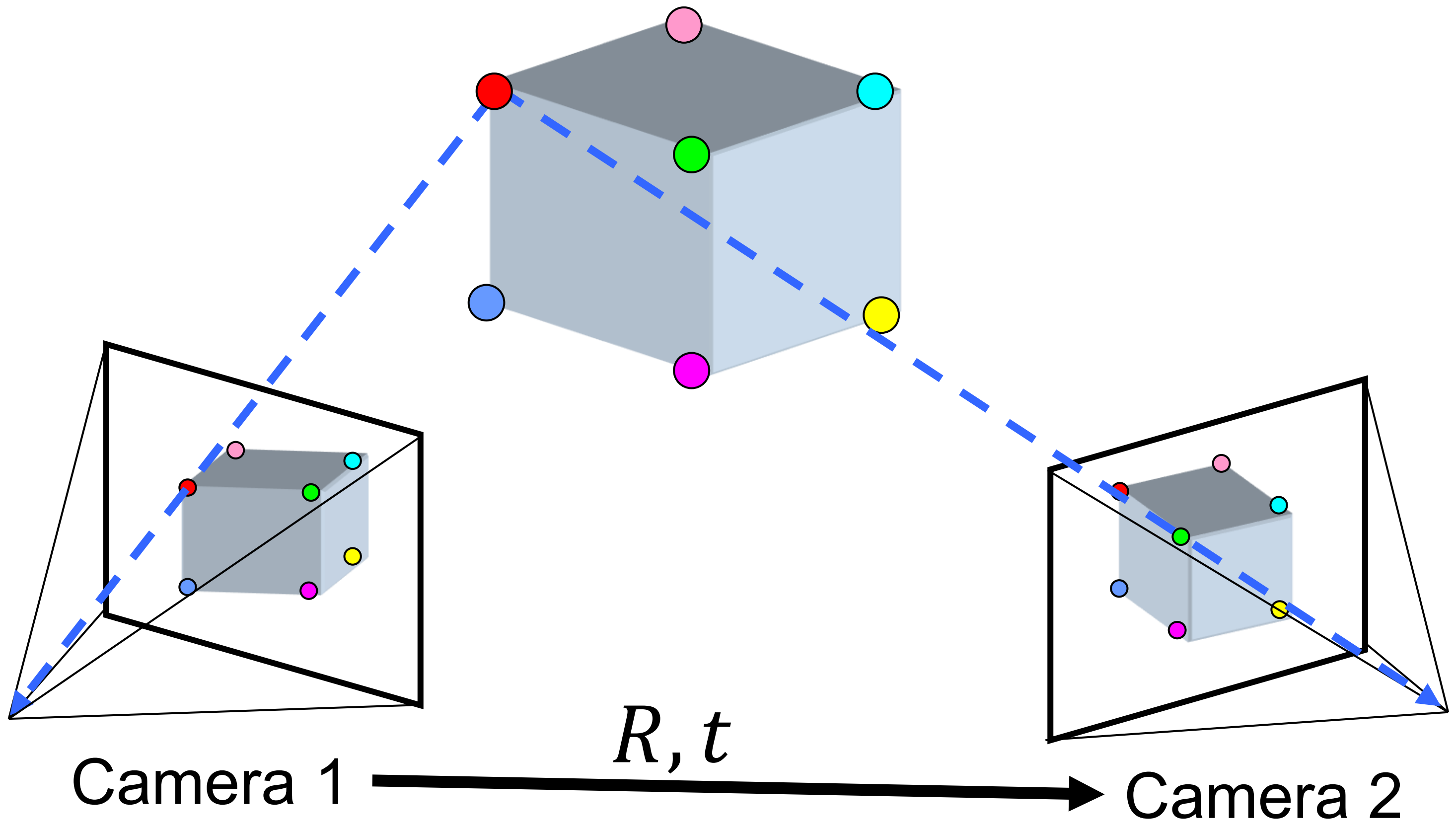
Ames room



- Single-view ambiguity:
  - Only know how a 3D point in world space corresponds to each pixel
  - Can only project 3D points to 2D points, process cannot reverse.
  - Nowhere near enough constraints to find 3D point from 2D point

Figure from EECS 442 University of Michigan, Credit: Prof. Justin Johnson

Resolving Single-view Ambiguity: Two-view Geometry

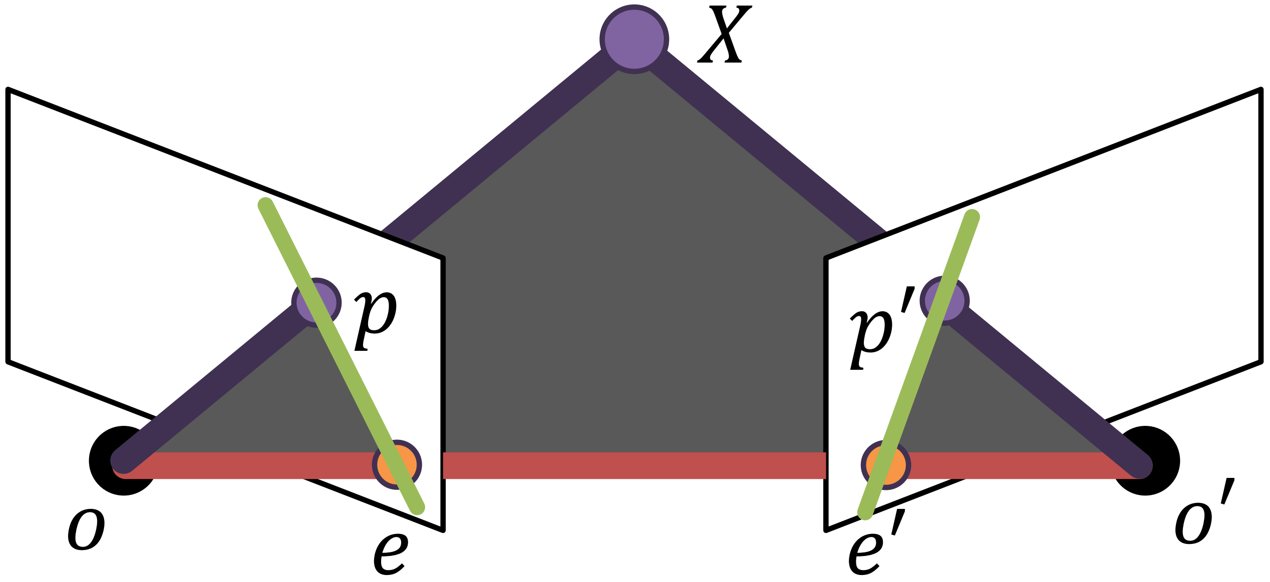


- Two-view geometry:
  - Stereo/Epipolar geomery:
  - Given 2 cameras with the extrinsic matrix (i.e., Rotation and Translation matrix) of Camera 2 with respect to Camera 1 known, find where a point could be in 3D space.
  - We start our discussion of two-view geometry with epipolar geometry.
  - The key: finding corresponding points in the two views.
  - Stereo geometry is analyzed under a special scenario where the two cameras are parallel to each other.

Figure from EECS 442 University of Michigan, Credit: Prof. Justin Johnson

Epipolar Geometry

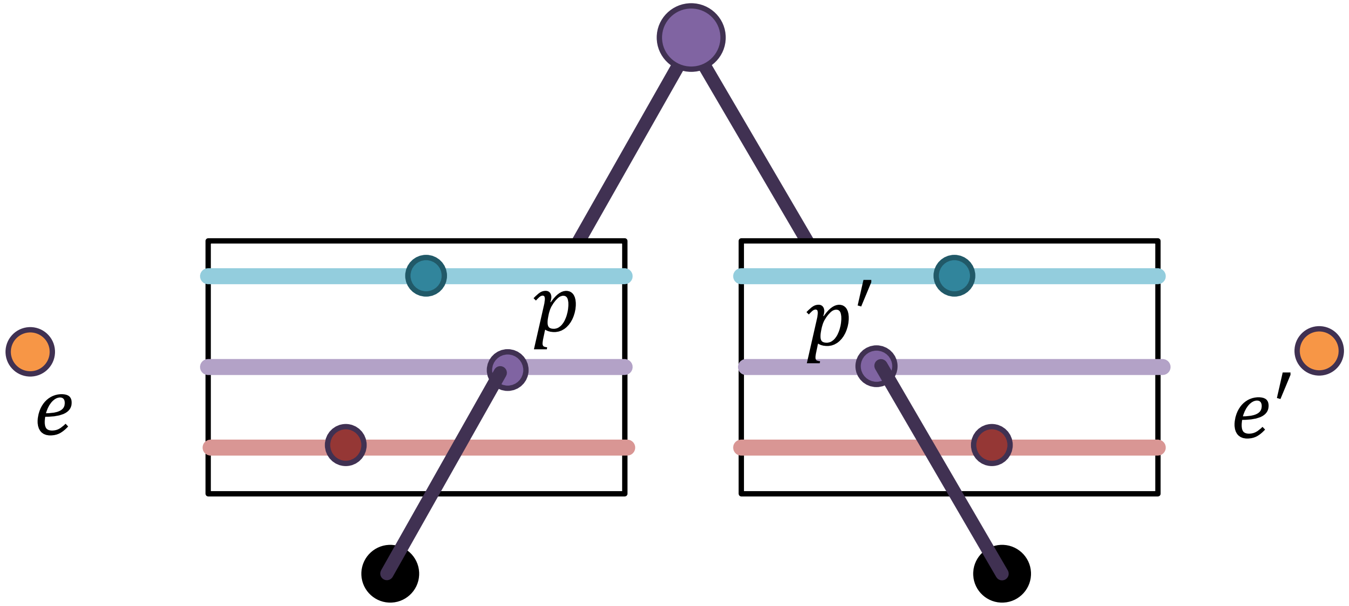
- Goal of analyzing epipolar geometry: finding the corresponding points across two views.



- Epipolar plane: the plane formed by the 3D point and the two camera centers.
- Baseline: the line that connects the two camera centers.
- Epipoles: where the baseline intersects the image planes: e and e'.
- Epipoles are essentially the projection of the other camera on the image plane.
- Epipolar lines are the lines that go between the epipoles and the projection of the points, and are the intersection of the epipolar plane with the image planes.
- The projection of the same object must fall onto the epipolar line.

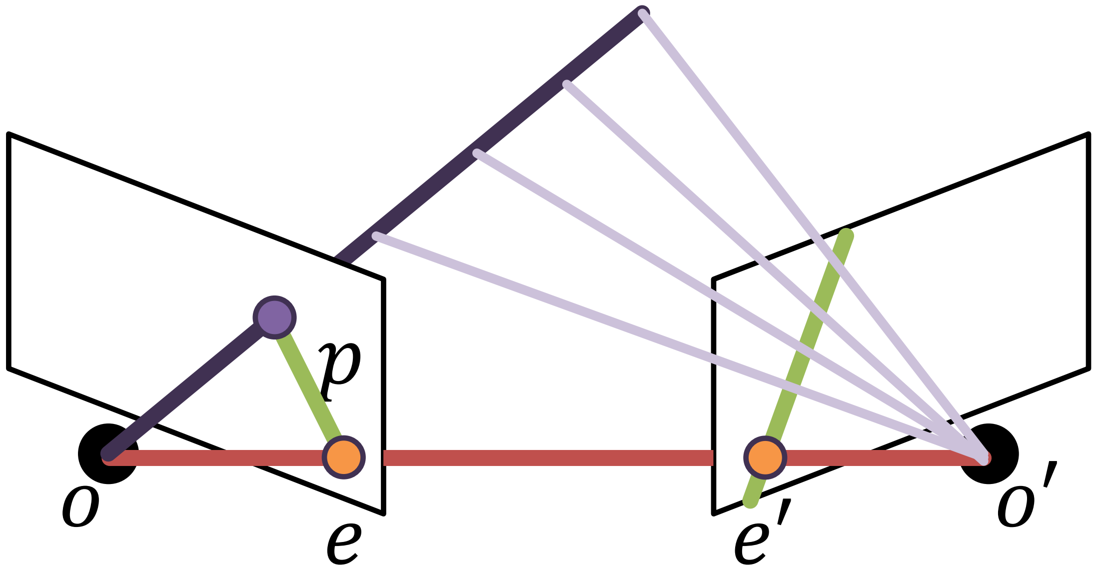
Figure from EECS 442 University of Michigan, Credit: Prof. Justin Johnson

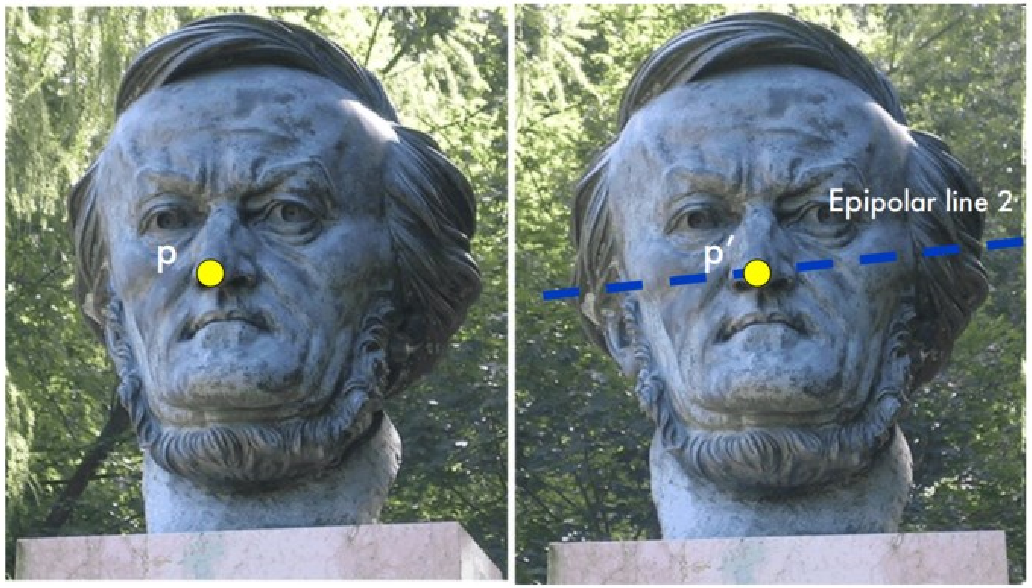
Epipolar Geometry: Cases when Two Image Planes are Parallel



- Under stereo view: the epipoles are *infinitely* far away while the epipolar lines are parallel to each other.

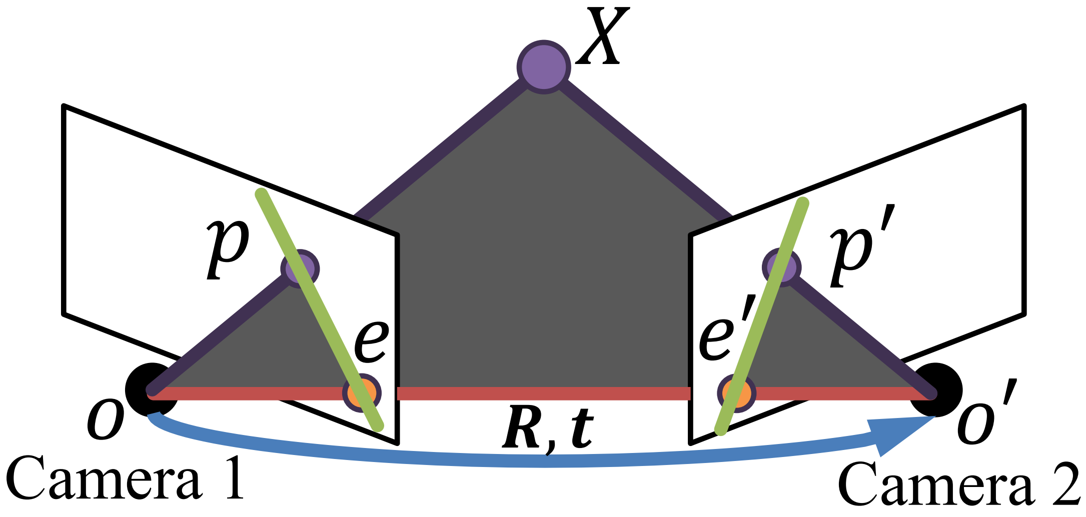
Epipolar Geometry: Epipolar Constraint





- Suppose we do not know the location of the point at the 3D space and only have p – the corresponding p' is on the corresponding epipolar line.
- Why analyze with epipolar geometry?
  - Naïve search: for each pixel, search every other pixel on the second view.
  - Search with epipolar geometry: for each pixel, seach ONLY along the epipolar line.
- Goal of epipolar geometry analysis: find the epipolar line given the camera model with intrinsic/extrinsic parameters.

Epipolar Constraint: Epipolar Line with Intrinsic/Extrinsic Params

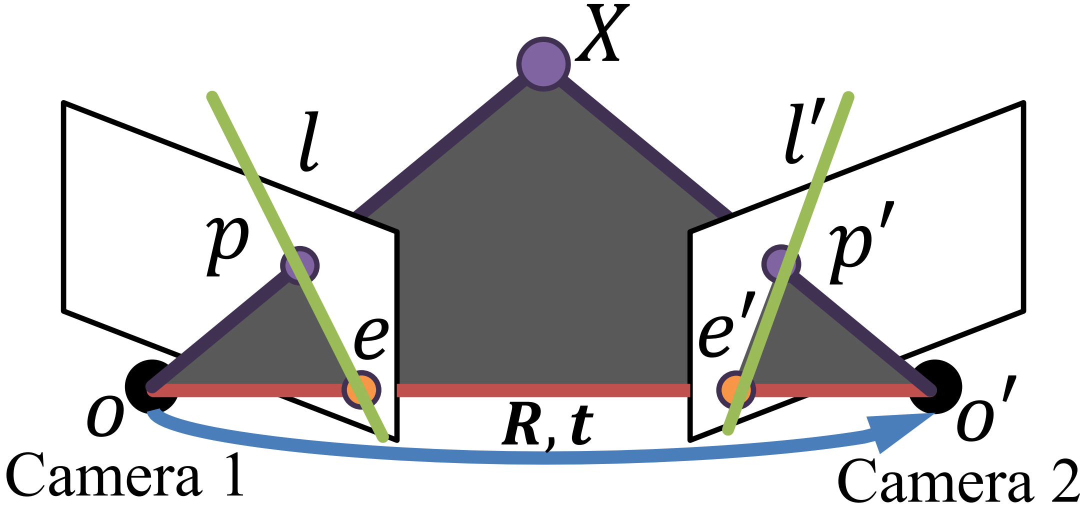


- Assume that we know the intrinsic and extrinsic parameters of the two cameras, while the coordinate system is set to be centered at Camera 1. The projection matrices (or the camera model matrices) are therefore denoted as:
- $\boldsymbol{M_1} = \boldsymbol{K}[\boldsymbol{I}, 0], \boldsymbol{M_2} = \boldsymbol{K}'[\boldsymbol{R}, \boldsymbol{t}]$
- Since we assume the intrinsic parameters $\boldsymbol{K}$ and $\boldsymbol{K}$' are known, we can obtain the normalized coordinates of the projected points p and p', denoted by: $\hat{p} = \boldsymbol{K}^{-1}p$ and $\widehat{p'} = \boldsymbol{K}'^{-1}p'$.
- Obviously, the following must be coplanar: $\hat{p}$, $\boldsymbol{t}$, and $\boldsymbol{R}\widehat{p'}$. Therefore, the co-planarity condition can be applied to the three vectors by: $\hat{p}^T\left(\boldsymbol{t} \times \boldsymbol{R}\widehat{p'}\right) = 0$.
- The triple product is mathematically equivalent to the following matrix multiplication:

$$\hat{p}^T\left(\boldsymbol{t} \times \boldsymbol{R}\widehat{p'}\right) = 0 \equiv \hat{p}^T \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix} \boldsymbol{R}\widehat{p'} = 0 \equiv \hat{p}^T [t_x] \boldsymbol{R}\widehat{p'} = 0$$

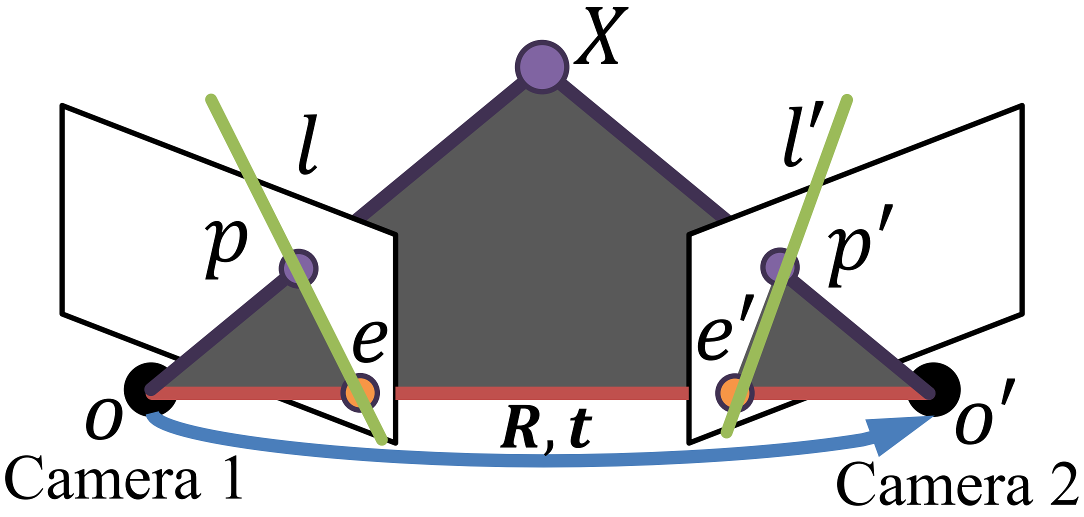
Epipolar Constraint: Epipolar Line with Intrinsic/Extrinsic Params

- With $\hat{p}^T[t_x]\boldsymbol{R}\widehat{p'} = 0$, we can define the essential matrix as $\boldsymbol{E} = [t_x]\boldsymbol{R}$, which leads to the following epipolar constraint with essential matrix:
- If you have a normalized point $\hat{p}$, its correspondence $\widehat{p'}$ **must** satisfy $\hat{p}^T\boldsymbol{E}\widehat{p'} = 0$.
- The epipolar lines are computed by:
- $l = \boldsymbol{E}\widehat{p'}$ and $l' = \boldsymbol{E}^T\hat{p}$.
- Further, all epipolar lines **must** pass through the epipoles, therefore the following two equation holds:
- $l^T\hat{e} = 0 \rightarrow \boldsymbol{E}^T\hat{e} = 0$ and $l'^T\widehat{e'} = 0 \rightarrow \boldsymbol{E}\widehat{e'} = 0$.



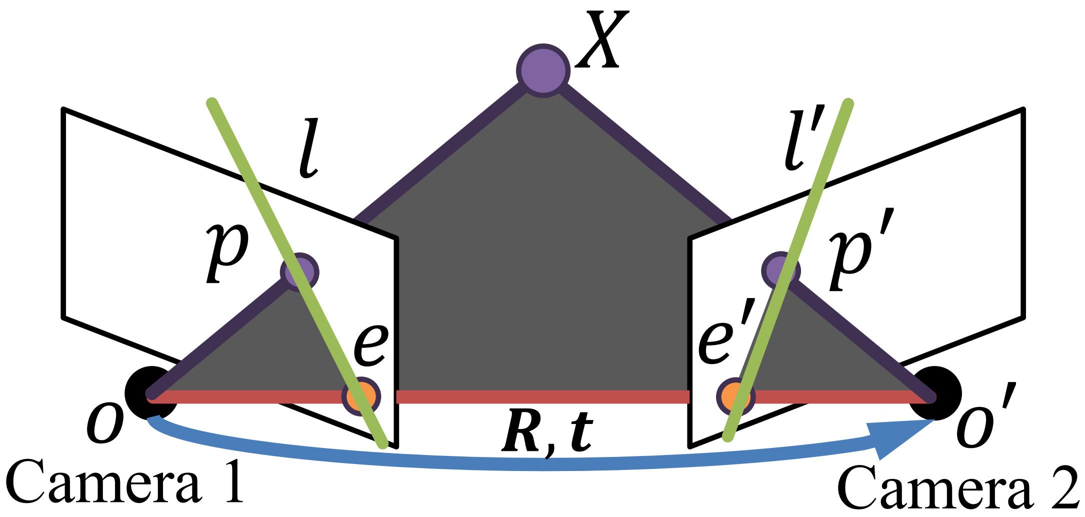
Epipolar Constraint: Notes on Epipolar Line

- We have $\hat{p}^T \boldsymbol{E} \widehat{p'} = 0$.
- Meanwhile, since $\hat{p}$ is on the epipolar line l, we therefore also have $l^T \hat{p} = 0 = \hat{p}^T l$.
- Thus the following equation stands:
  - $l = \boldsymbol{E} \widehat{p'}$.
- Vice versa for epipolar line l' and $\widehat{p'}$ where we have $l'^T \widehat{p'} = 0$, and therefore $l'^T = \hat{p}^T \boldsymbol{E}$, which leads to $l' = \boldsymbol{E}^T \hat{p}$.



Epipolar Constraint: Epipolar Line with Extrinsic Params

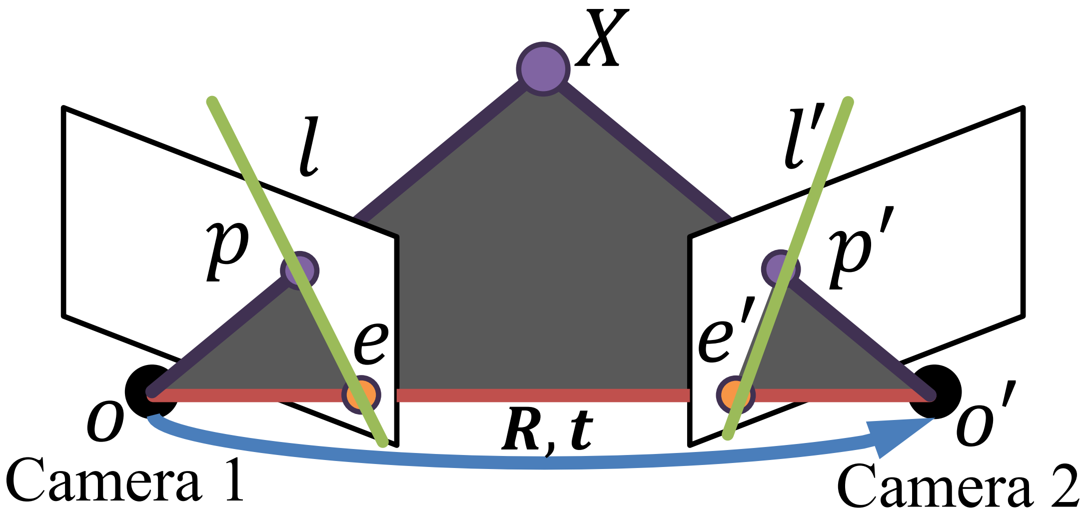
- The epipolar constraint with essential matrix can only be reached when the intrinsic parameters are known. In reality, this may **NOT** be the case.
- What if we do **NOT** know the intrinsic parameters but **ONLY** the extrinsic parameters of Camera 2 with respect to Camera 1?



- Recap: $\hat{p} = \boldsymbol{K}^{-1}p$, $\widehat{p'} = \boldsymbol{K}'^{-1}p'$, and $\hat{p}^T\boldsymbol{E}\widehat{p'} = 0$.
- Since the intrinsic parameters are unknown, the normalized coordinates cannot be computed. We reverse to the original coordinate on the image plane by sub in $\hat{p} = \boldsymbol{K}^{-1}p$, $\widehat{p'} = \boldsymbol{K}'^{-1}p'$:
- $(\boldsymbol{K}^{-1}p)^T\boldsymbol{E}\left(\boldsymbol{K}'^{-1}p'\right) = 0 \equiv p^T\boldsymbol{K}^{-T}\boldsymbol{E}\boldsymbol{K}'^{-1}p' = 0.$
- We define the fundamental matrix as $\boldsymbol{F} = \boldsymbol{K}^{-T}\boldsymbol{E}\boldsymbol{K}'^{-1}$, we can therefore obtain the epipolar constraint with the fundamental matrix as: $p^T\boldsymbol{F}p' = 0$.

Epipolar Constraint: Epipolar Line with Extrinsic Params

- Similar to the epipolar constraint with essential matrix, the epipolar lines are computed by:
- $l = \boldsymbol{F}p'$ and $l' = \boldsymbol{F}^T p$
- Further, all epipolar lines **must** pass through the epipoles, therefore the following two equation holds:
- $l^T e = 0 \rightarrow \boldsymbol{F}^T e = 0$ and ${l'}^T e' = 0 \rightarrow \boldsymbol{F} e' = 0.$
- The fundamental matrix $\boldsymbol{F}$ is of 7 degrees of freedom (DOF) – estimated by an 8-point algorithm (use 8 points).
- For those interested, refer to https://www.cs.cmu.edu/~16385/s17/Slides/12.4_8Point_Algorithm.pdf



Epipolar Constraint: Estimating $\boldsymbol{F}$ with Normalization

- Notes on the 8-point algorithms: solving $\boldsymbol{F}$ by $[u, v, 1]\begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix}\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = 0$
- 1. Compute with homogeneous coordinates.
- 2. Using the original projected image point values may induce large-scale figures during matrix multiplication (up to 10^{12} scale for a 1000×1000 image).
- To solve the 2nd problem: **normalization** – the matrix computed becomes better conditioned.
- Normalization applied: image origin becomes the centroid of the image plane (zero-meaned) while the mean squared distance between the origin and the data points is scaled to 2 pixels (scaled-variance).

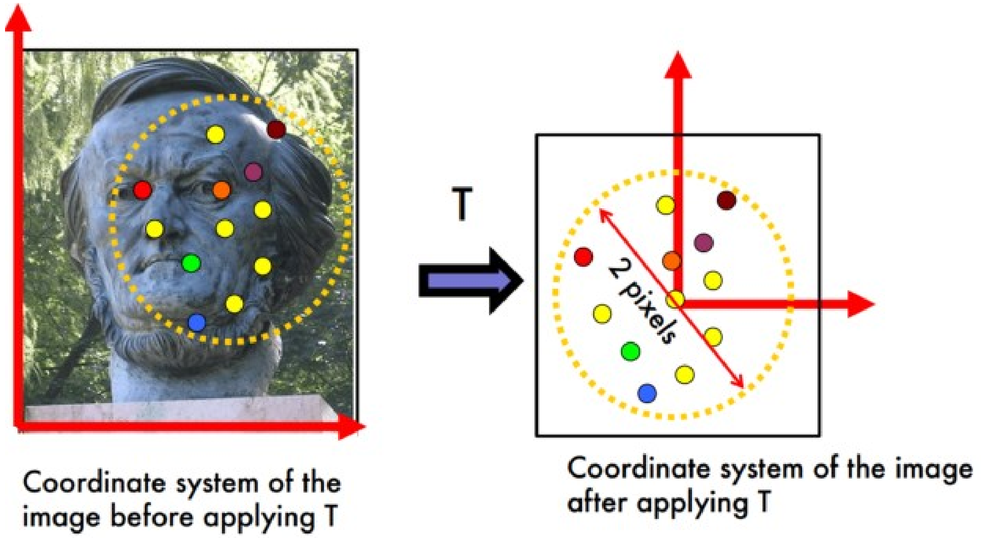
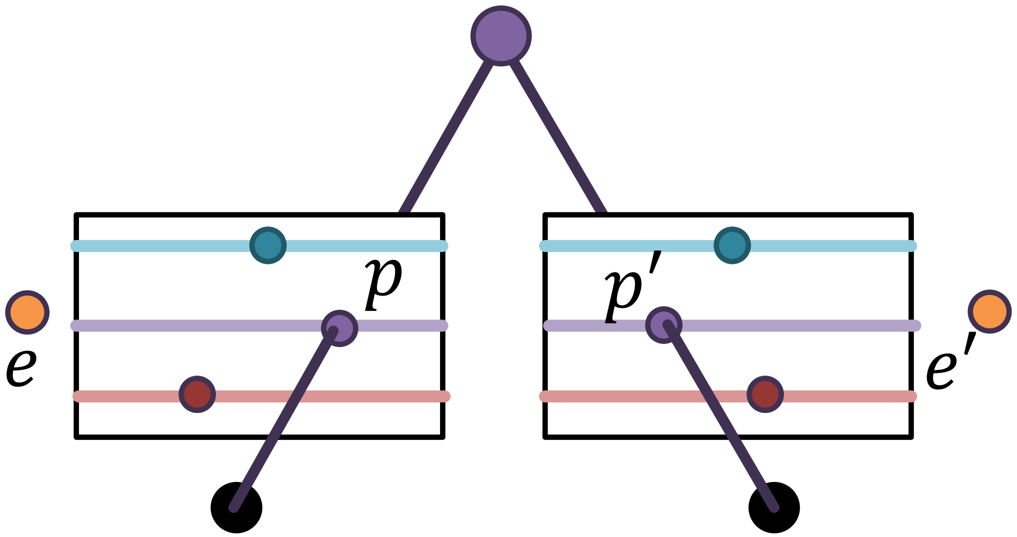
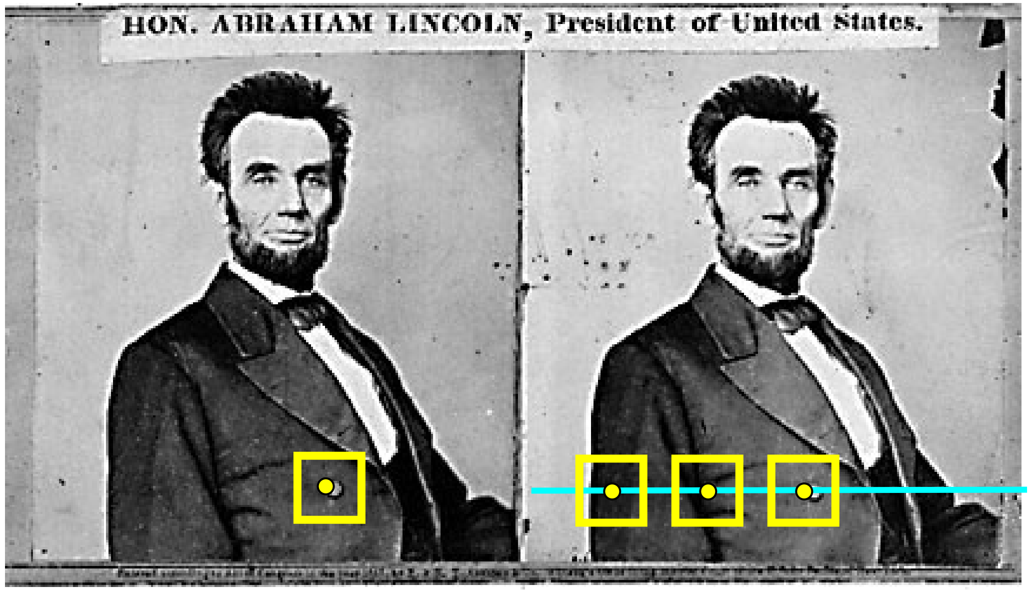


Figure from CS231A Stanford University, Credit: Prof. Silvio Savarese

Stereo Geometry: Cases when Two Image Planes are Parallel



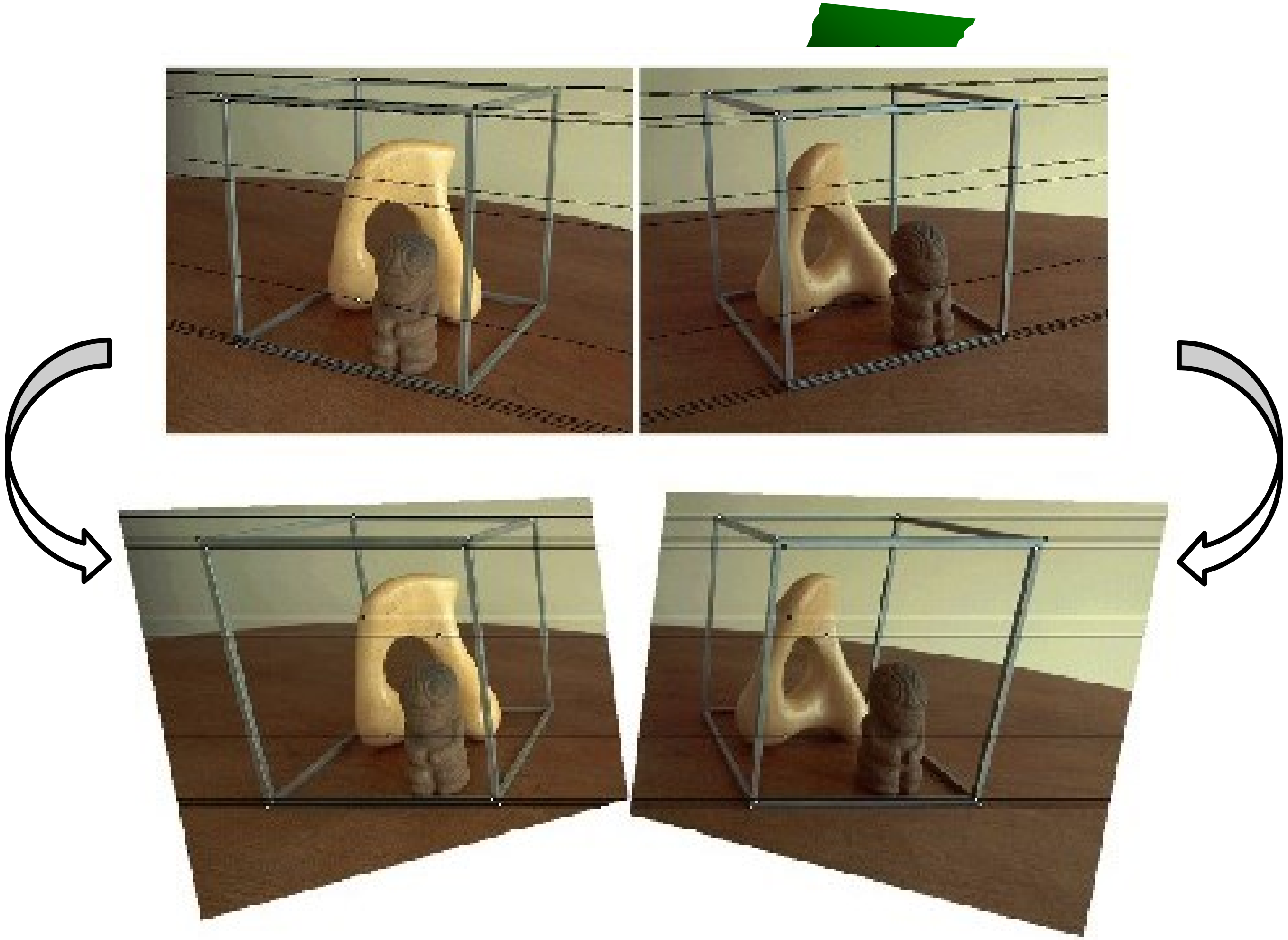


- Now lets consider the more special case – this case is similar to how humans perceive the environment through stereo vision.
- In this scenario, image planes of cameras are parallel to each other and to the baseline, while the camera centers are set at same height, the focal lengths the same → epipolar lines fall along the horizontal scan lines of the images.
- In this scenario, R=I, and t=[T,0,0] (shift along x-axis). We do not need to find the epipolar line anymore.
- The focus of stereo geometry thus changes to obtaining the **depth information** by triangulating the matches.
- → How humans can fuse pair of images from the left and right eyes to get a **sensation** of depth.

Figure from EECS 442 University of Michigan, Credit: Prof. Justin Johnson

Stereo Geometry: Stereo Image Retification

- As the case of parallel, same height images are much easier to analyze and to find the corresponding pixels for subsequent tasks such as 3D reconstruction, how can we tune any two-view images to match the conditions of analyzing through stereo geometry?
- Stereo image rectification: reproject image planes onto a common plane parallel to the line between optical centers.



Stereo Geometry: Depth from Disparity

- From similar triangles, the following two equation holds:

$$\frac{x}{f} = \frac{B_1}{z}; \frac{-x'}{f} = \frac{B_2}{z}$$

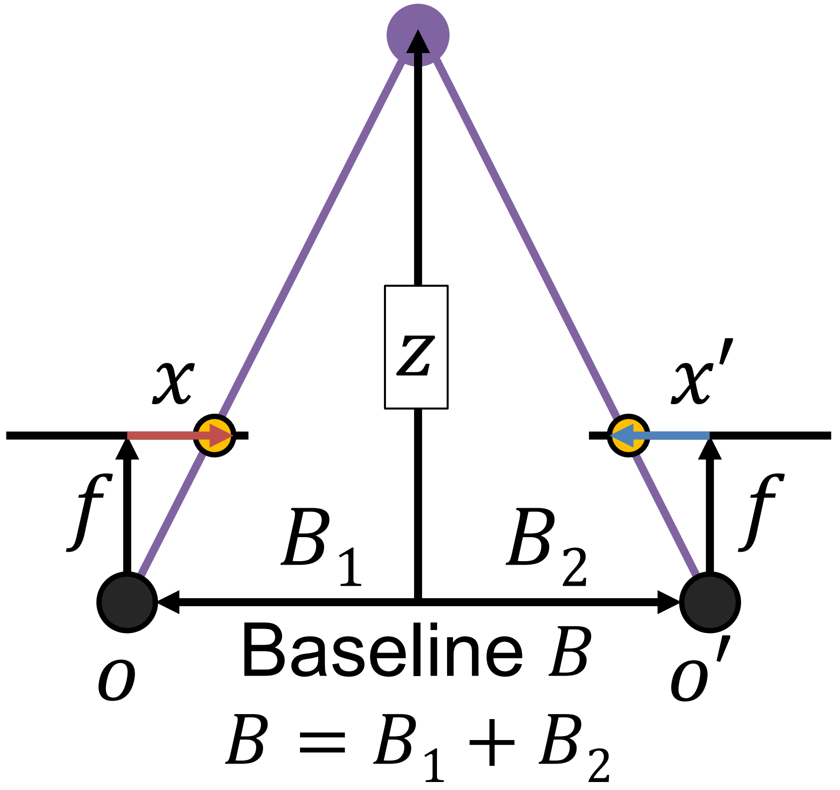
- We add the two equations and obtain:

$$\frac{x - x'}{f} = \frac{B_1 + B_2}{z} = \frac{B}{z}$$

- Denote the disparity as the distance between two corresponding points in the left and right image of a stereo pair, that is $d = x - x'$, we can therefore obtain the depth from disparity by:

$$d = x - x' = \frac{fB}{z}$$

$$z = \frac{fB}{d}$$



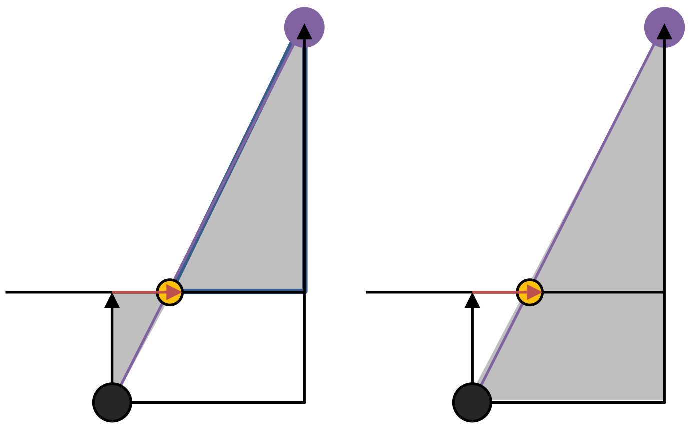


Figure from EECS 442 University of Michigan, Credit: Prof. Justin Johnson

Stereo Geometry: Depth from Disparity

- Denote the disparity as the distance between two corresponding points in the left and right image of a stereo pair, that is $d = x - x'$, we can therefore obtain the depth from disparity by:

$$d = x - x' = \frac{fB}{z}, z = \frac{fB}{d}$$

- This holds whether the object of interest is projected on the same side of the image planes or on different sides (see last slide)

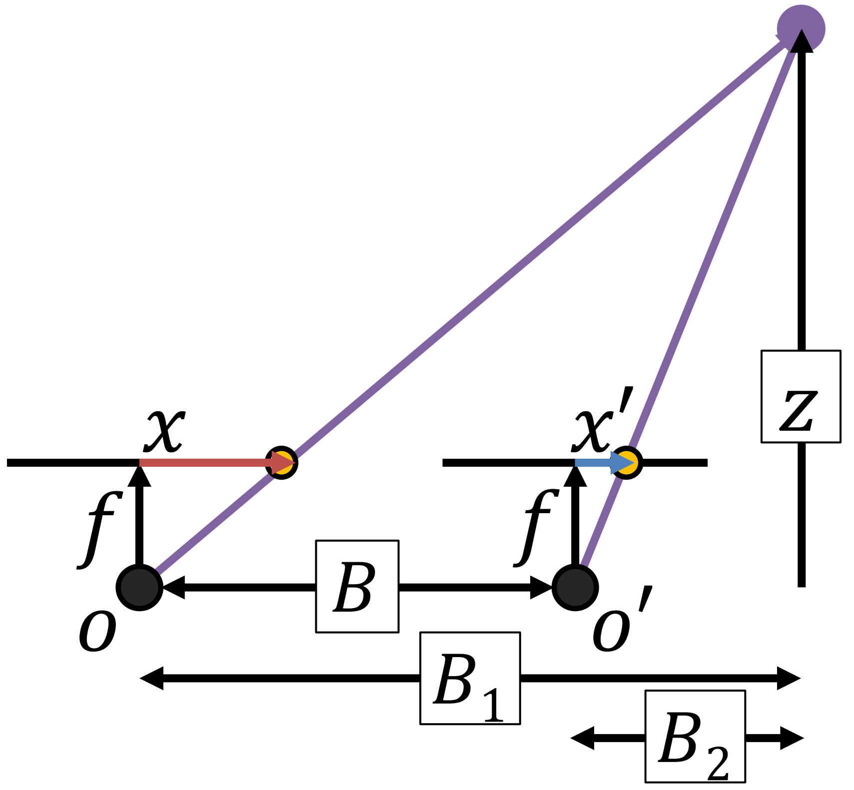


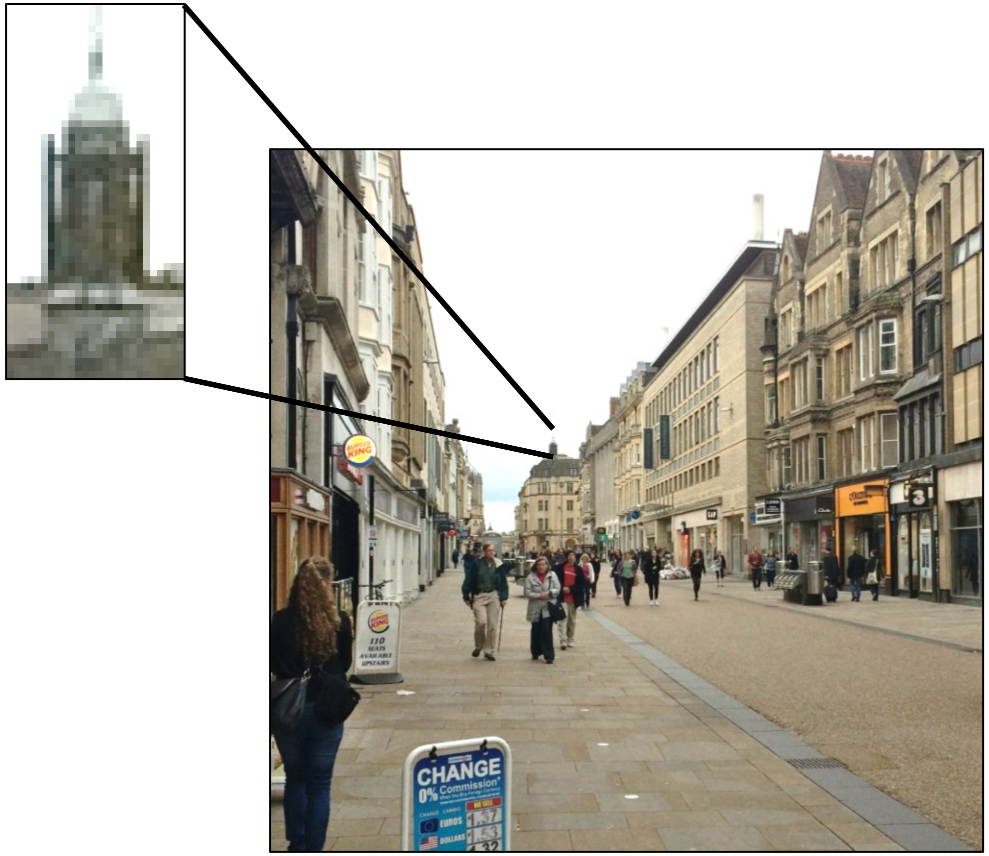
Figure from EECS 442 University of Michigan, Credit: Prof. Justin Johnson

Stereo Geometry: Depth Error from Disparity Error

- Depth estimation – disparity measurements may not be correct due to blurry images/low resolution.
- What is the effect of disparity error towards depth estimation? Expressing depth error from disparity error.
- Assume that we may not know the exact disparity d, but the estimated depth Z, focal length f, baseline length B and disparity error Δd are known.
- To estimate ΔZ, we start from the depth estimation:

$$Z = \frac{fB}{d}$$

$$\Delta Z = \frac{fB}{d} - \frac{fB}{d + \Delta d} = fB \frac{\Delta d}{d(d + \Delta d)} = \frac{Z^2 \, \Delta d}{fB + Z\Delta d}$$



Stereo Geometry: Depth Error from Disparity Error

$$\Delta Z = \frac{fB}{d} - \frac{fB}{d+\Delta d} = fB\frac{\Delta d}{d(d+\Delta d)} = \frac{Z^2\,\Delta d}{fB + Z\Delta d}$$

- However, this is not the simplest form, $\Delta Z = g(\Delta d)$ is not a linear function.
- To simplify, we use first order Taylor series estimation in the form of: $g(x)|_{x\to a} = g(a) + g'(x)|_{x=a}(x-a)$
- Therefore (***important open question: how?***), the depth error is estimated from the disparity error as: (Tip: $g(\Delta d) = g(0+\Delta d)$)

$$\Delta Z = \frac{Z^2}{fB}\Delta d$$

- Implication: the further the object is in 3D space, the more uncertain it is in depth computation – MUCH larger depth estimation.