Università di Genova | DIBRIS DIPARTIMENTO DI INFORMATICA, BIOINGEGNERIA, ROBOTICA E INGEGNERIA DEI SISTEMI

Augmented Reality

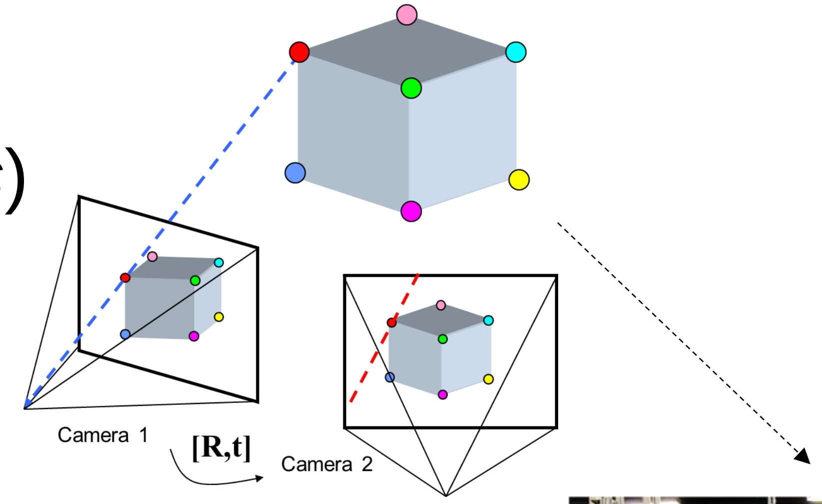
Lecture 11 – stereo vision and 3D reconstruction

Manuela Chessa – manuela.chessa@unige.it

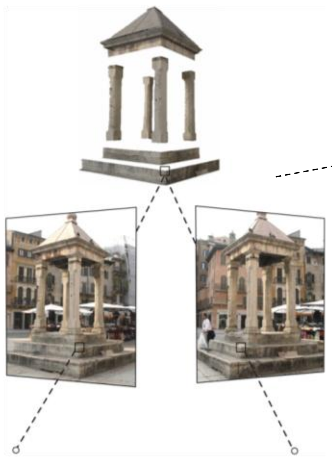
Fabio Solari – fabio.solari@unige.it

Summary

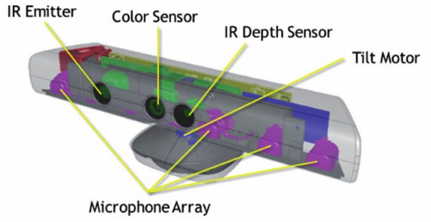
- Stereo Vision (*Geometry of two views*)
  - Stereopsis and epipolar geometry
  - Essential matrix
  - Fundamental matrix
  - Eight-point algorithm



- 3D reconstruction
  - Intrinsic and extrinsic parameters known
  - Only intrinsic parameters known



- RGB-D cameras and stereo displays

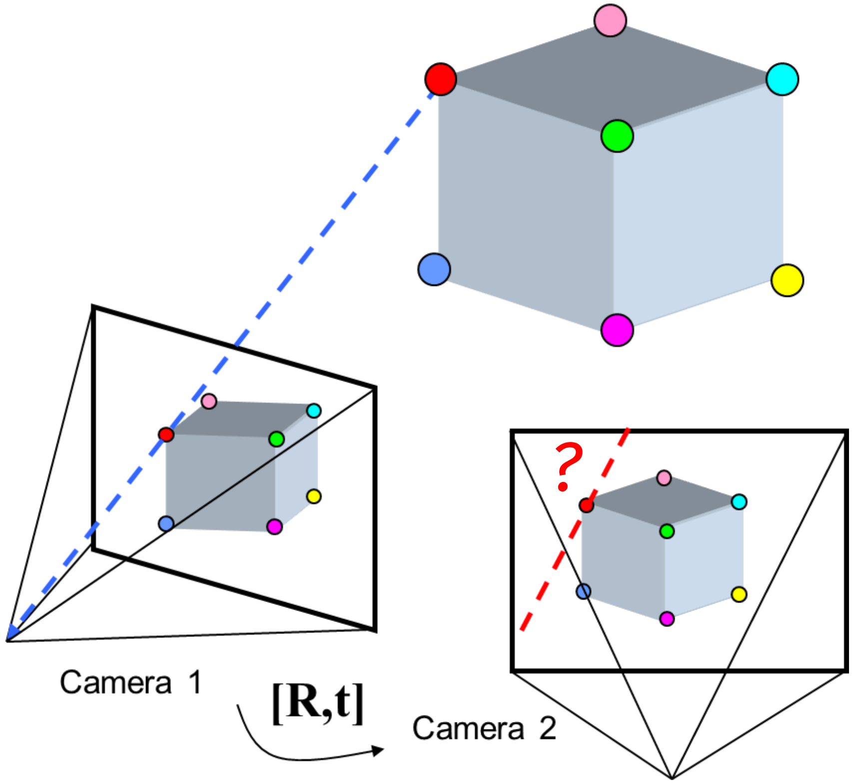


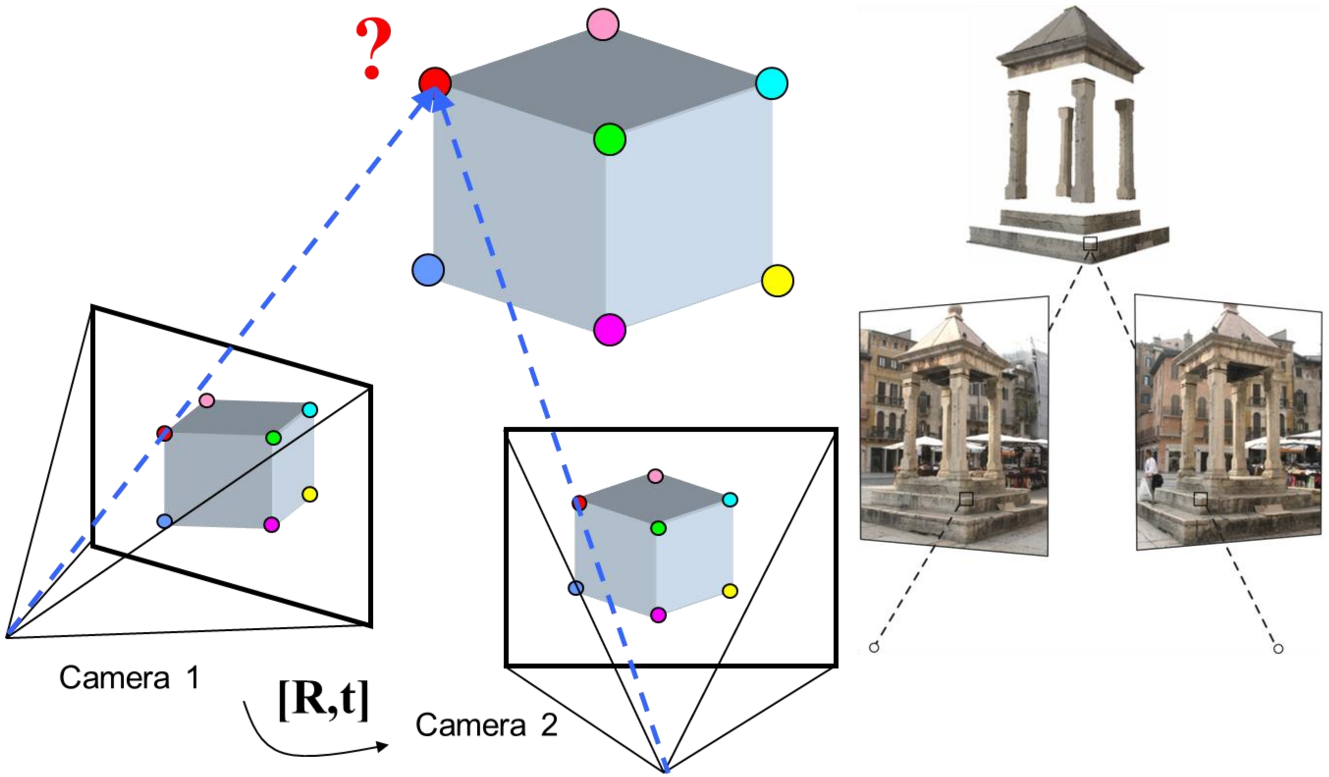
Università di Genova
DIBRIS DIPARTIMENTO
DI INFORMATICA, BIOINGEGNERIA,
ROBOTICA E INGEGNERIA DEI SISTEMI

Stereo Vision and epipolar geometry

Stereopsis problems (geometry of two views)

- **Stereo correspondence:** Given a point in one of the images, where could its corresponding points be in the other images (epipolar line)?
- **3D reconstruction:** Given the projections of the same 3D point in two (*or more*) images, compute the 3D coordinates of that point.

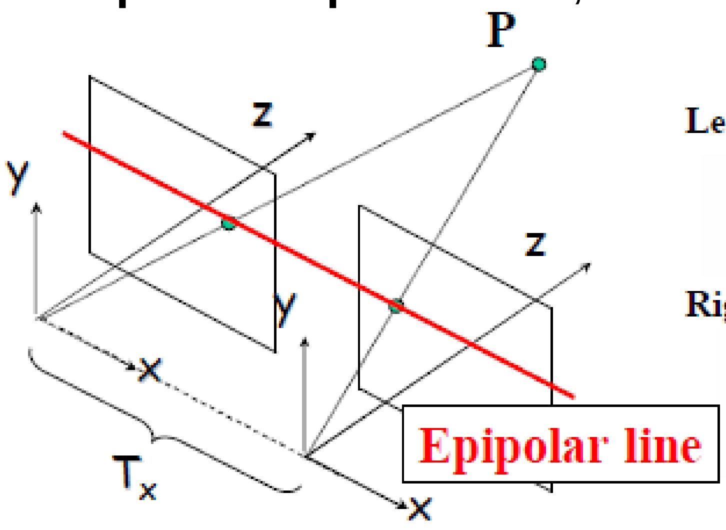




Here, we introduce the basic building blocks of the **geometry of two views**, known as ***epipolar geometry***, for the general case.

Epipolar geometry

Simple stereo system with **parallel optical axes**, Z axis (see previous lecture)



Left camera

$$x_l = f\frac{X}{Z} \qquad \boxed{y_l = f\frac{Y}{Z}}$$

Right camera

$$x_r = f\frac{X - T_x}{Z} \qquad \boxed{y_r = f\frac{Y}{Z}}$$

Same Y Coord!

depth

baseline

$$Z = \frac{f T_x}{d}$$

disparity

Equation relating depth and disparity

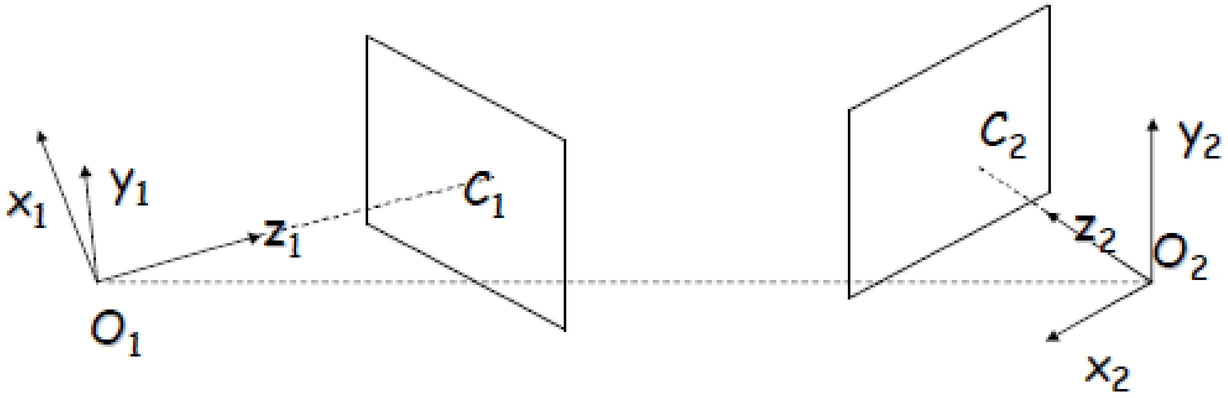
Important Stereo Vision Concept:

Given a point in the left image, we don't have to search the whole right image for a corresponding point.

The "epipolar constraint" reduces the **search space** to a one-dimensional **line**.

Epipolar geometry

General Stereo system



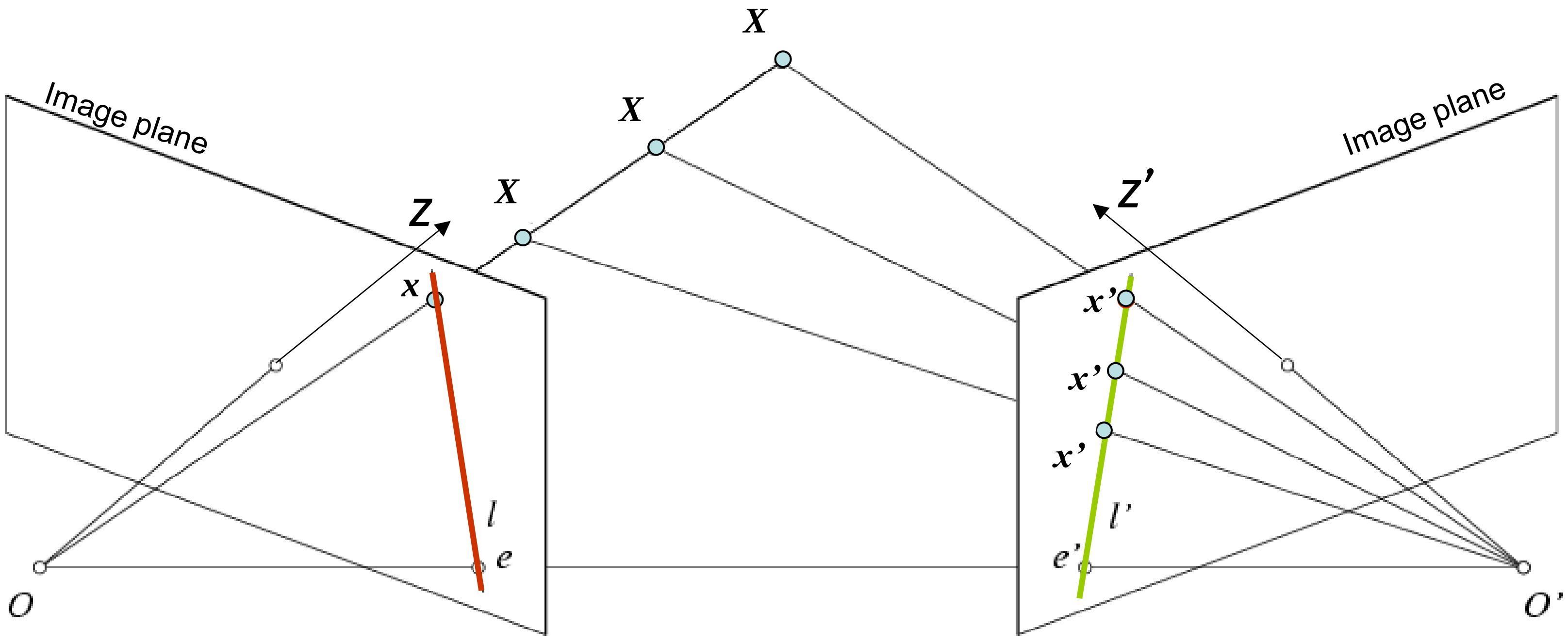
The **two calibrated cameras** are related by an arbitrary transformation (R,T):
*the **essential matrix** describes the relationships between the cameras*.

The **intrinsic** parameters of the two cameras are **unknown**:
***the fundamental matrix** describes the relationships between the cameras*.

The essential and fundamental matrices are 3x3 matrices that "encode" the epipolar geometry of two views.
Motivation: Given a point in one image, multiplying by the essential/fundamental matrix will tell us which **epipolar line** to search along in the second view.

Key idea: Epipolar constraint

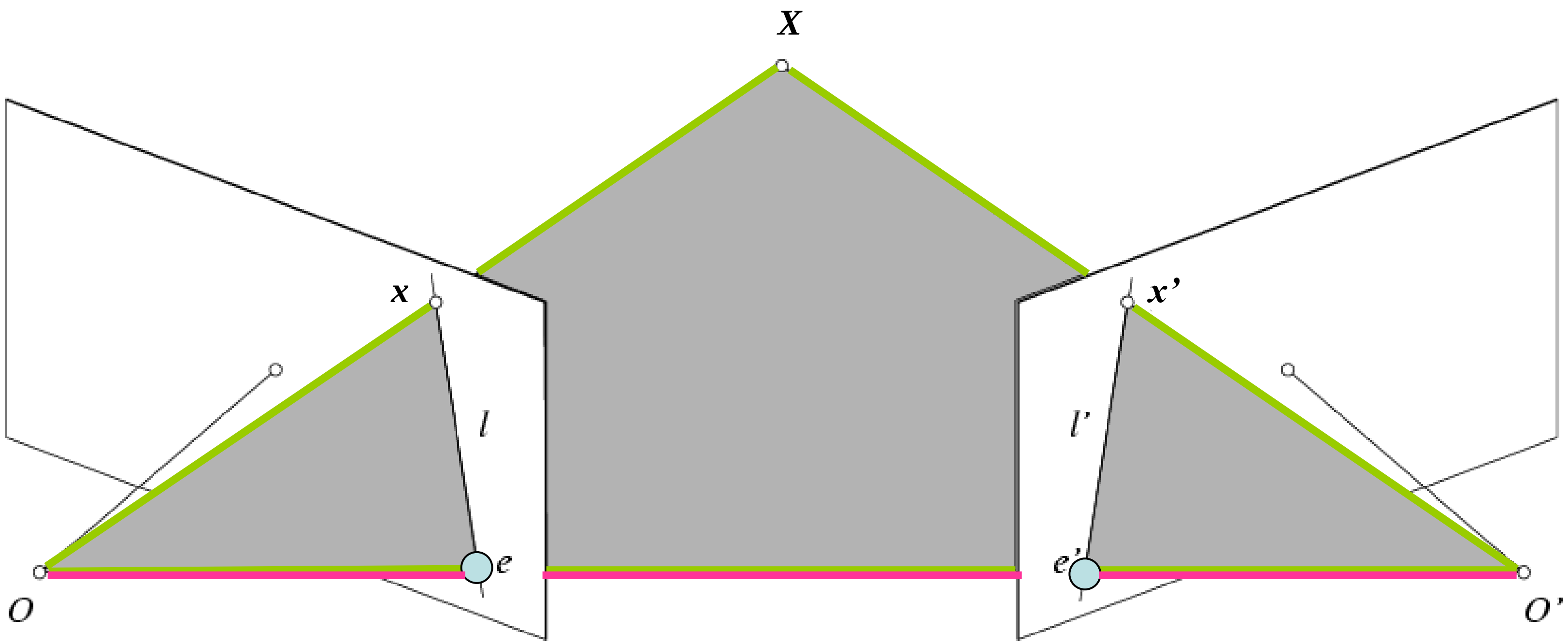
- It has been long known in photogrammetry that the coordinates of the **projection** (x,x') of a world **point** (X) and the two camera **optical centers** (O,O') form a **triangle**, a fact that can be written as an algebraic constraint involving the camera poses and image coordinates.



Potential matches for *x* must lie on the corresponding line *l'*.

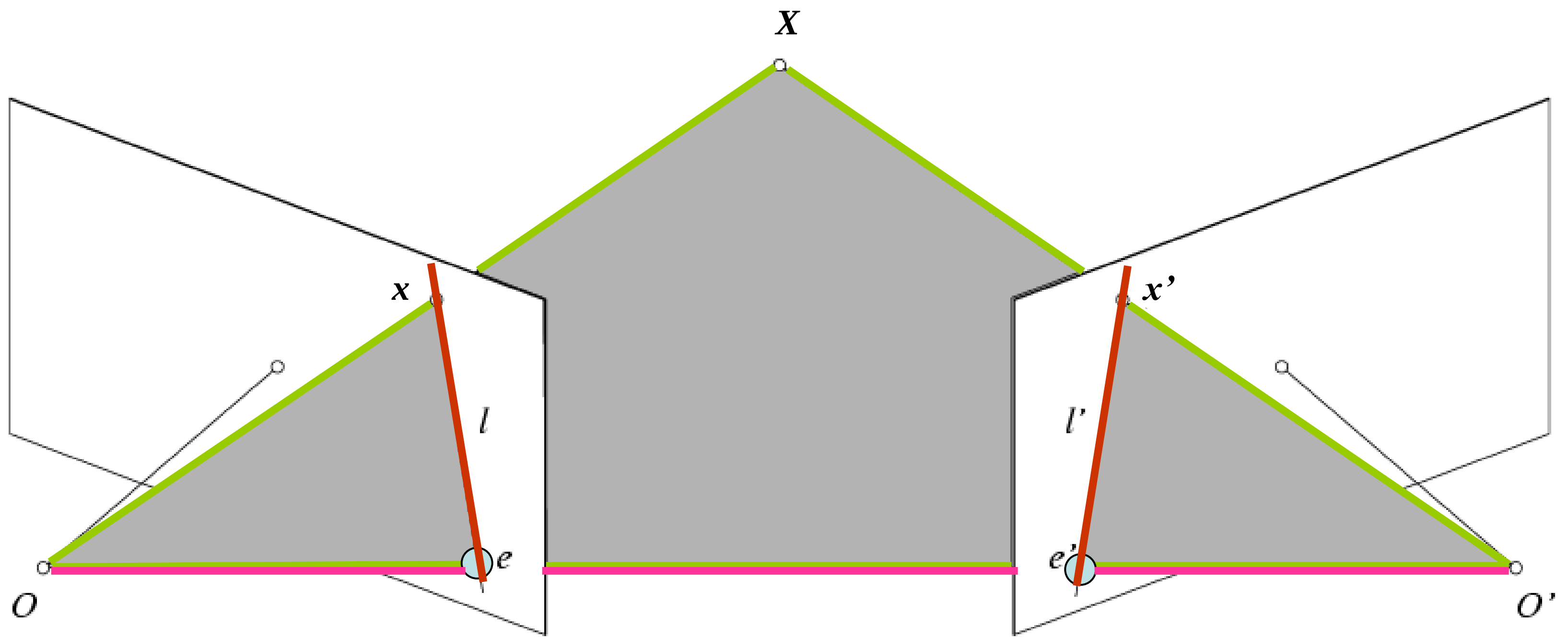
Potential matches for *x'* must lie on the corresponding line *l*.

Epipolar geometry: notation



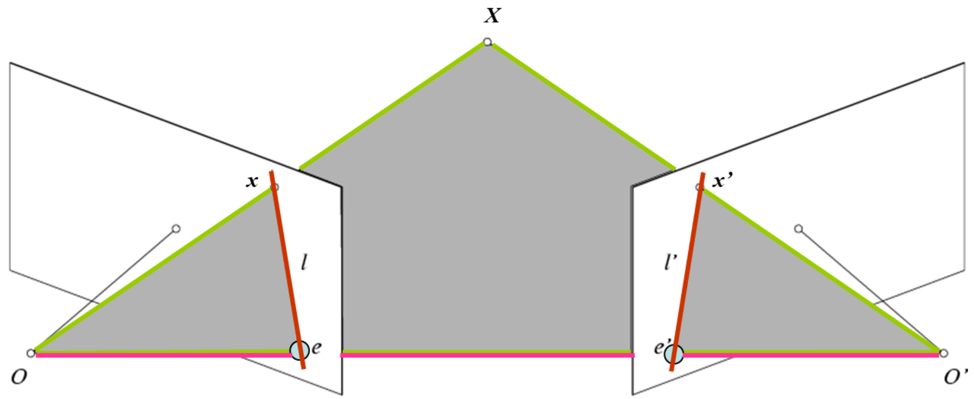
- **Baseline** – line connecting the two camera centers O and O`
- **Epipoles**
  - intersections of the baseline with image planes, e and e`
  - projections of the other camera center
- **Epipolar Plane** – plane containing baseline (1D family)

Epipolar geometry: notation



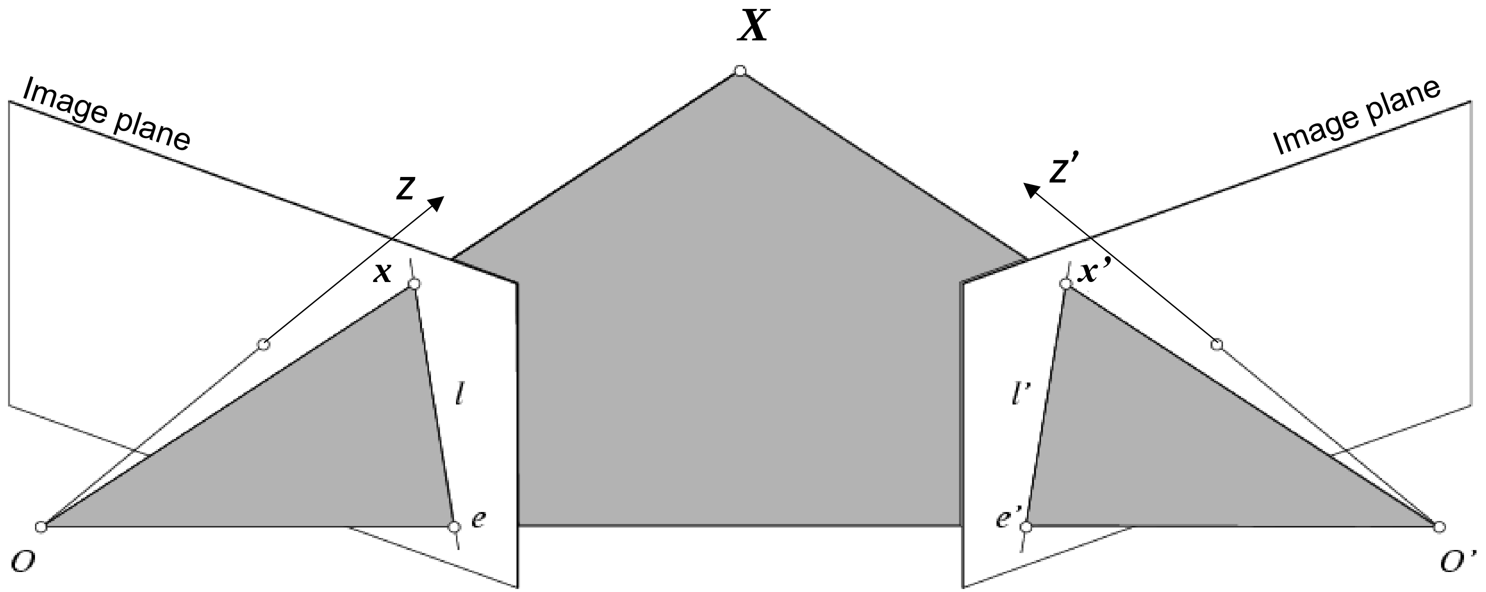
- **Baseline** – line connecting the two camera centers O and O\`
- **Epipoles**
  - intersections of the baseline with image planes, e and e\`
  - projections of the other camera center
- **Epipolar Plane** – plane containing baseline (1D family)
- **Epipolar Lines** - intersections of epipolar plane with image planes (always come in corresponding pairs), l and l\`

What is this useful for?



- **Find *x'***: If we know *x*, we can restrict *x'* to be along the line *l'*: **compute disparity** of stereo images (to find *x'*).
- Given candidate *x* and *x'* correspondences, estimate relative position and orientation between the cameras (**camera pose**).
- **Model fitting**: see if candidate *x*, *x'* correspondences fit estimated **projection models** of cameras 1 and 2.
- Given candidate *x* and *x'* correspondences, and having **calibrated cameras** (*known intrinsic K, K' and extrinsic relationship*), estimate the **3D position** of corresponding image points (**3D reconstruction**).

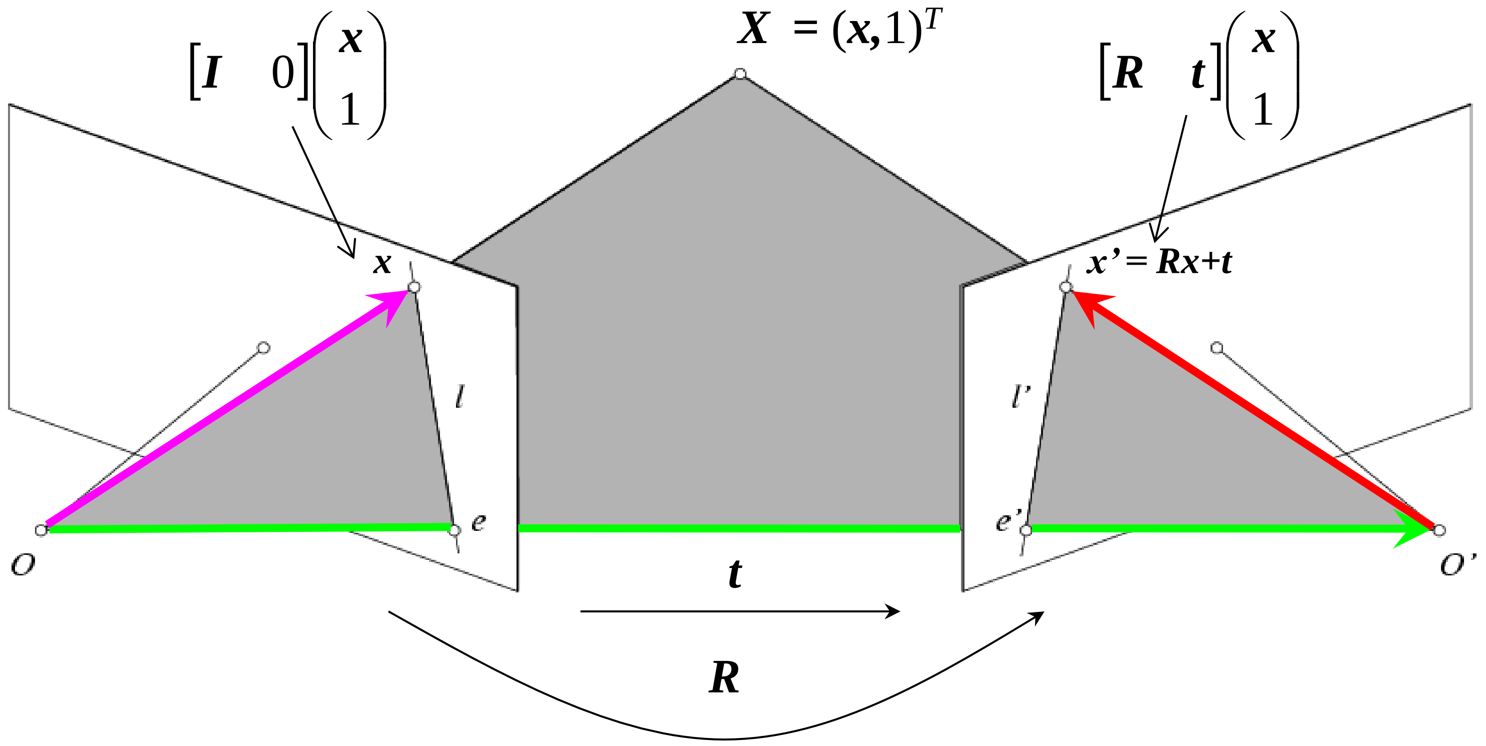
Epipolar constraint: Calibrated case



- **Intrinsic** and **extrinsic** parameters of the cameras are **known**, world coordinate system is set to that of the first camera
- Then the projection matrices are given by $\boldsymbol{K}[\boldsymbol{I} \mid \boldsymbol{0}]$ and $\boldsymbol{K}'[\boldsymbol{R} \mid \boldsymbol{t}]$
- We can multiply the projection matrices (and the image points) by the inverse of the calibration matrices to get *normalized* (**metric**) image coordinates:

$$\boldsymbol{x}_{\text{norm}} = \boldsymbol{K}^{-1}\boldsymbol{x}_{\text{pixel}} = [\boldsymbol{I} \;\; 0]\boldsymbol{X}, \qquad \boldsymbol{x}'_{\text{norm}} = \boldsymbol{K}'^{-1}\boldsymbol{x}'_{\text{pixel}} = [\boldsymbol{R} \;\; \boldsymbol{t}]\boldsymbol{X}$$

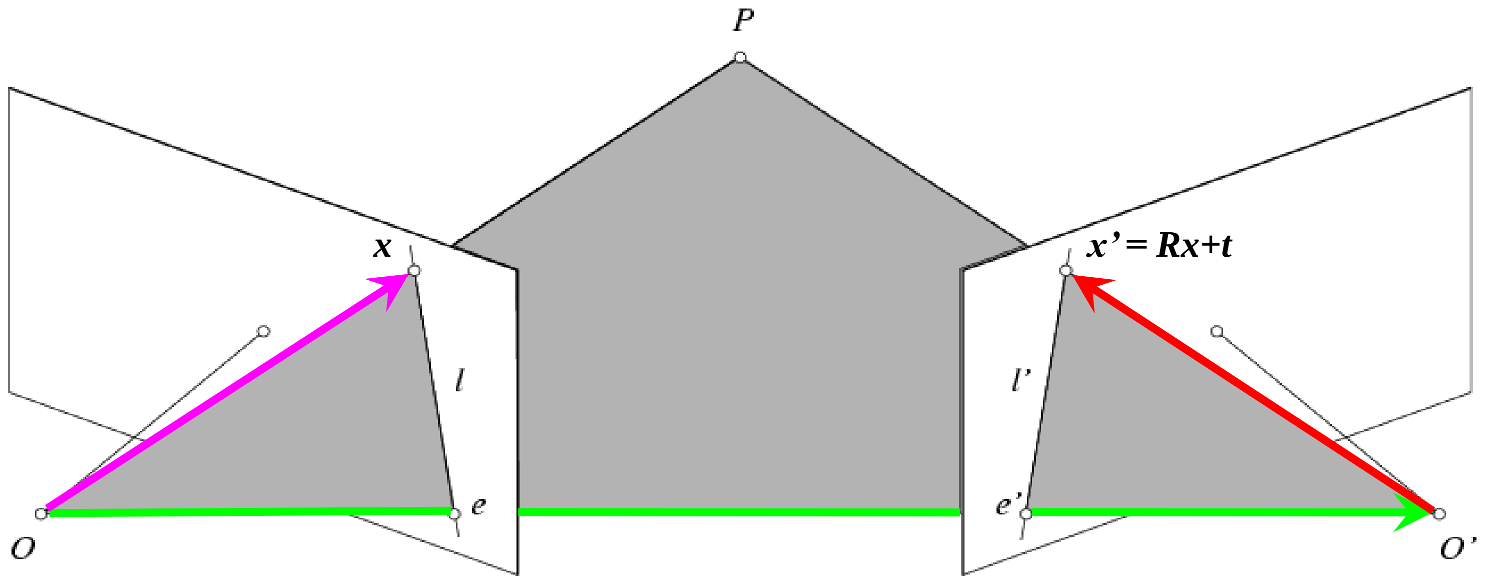
Epipolar constraint: Calibrated case



The vectors $\mathbf{Rx}$, $\mathbf{t}$, and $\mathbf{x'}$ are coplanar

X' is X in the second camera's coordinate system
We can identify the non-homogeneous 3D vectors X and X' with the homogeneous coordinate vectors x and x' of the projections of the two points into the two respective images

Epipolar constraint: Calibrated case



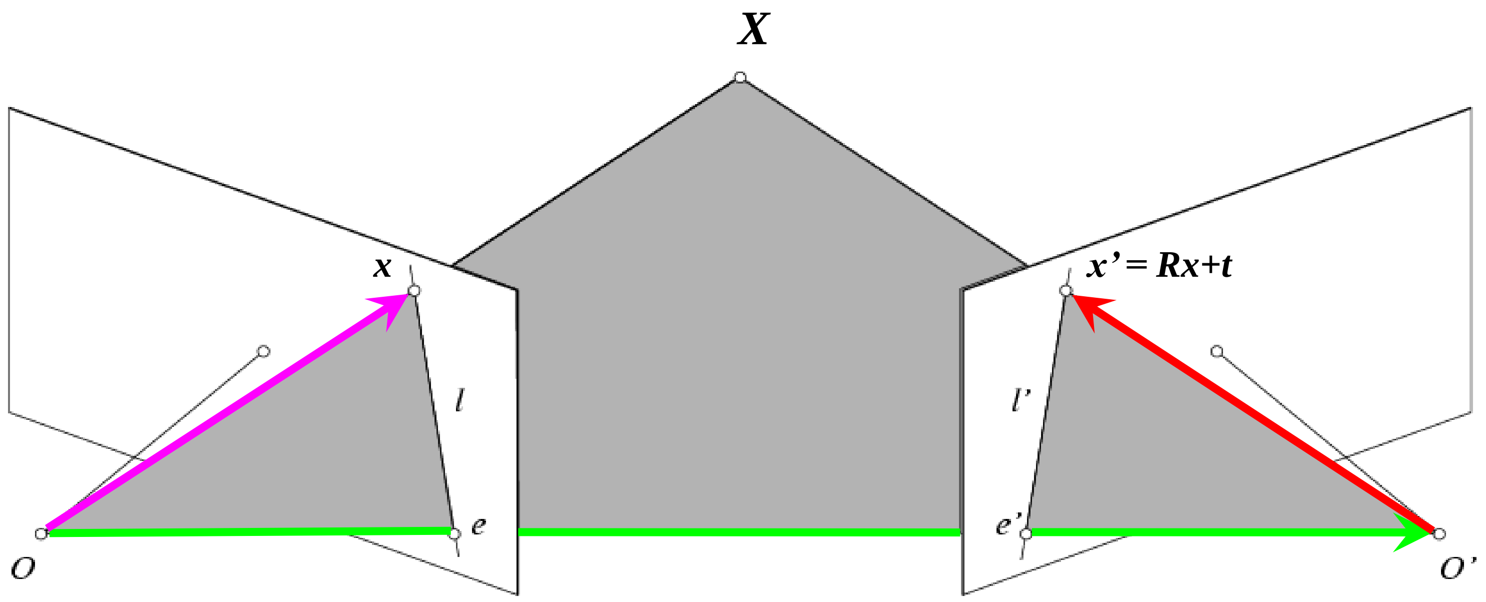
$$\boldsymbol{x}' \cdot [\boldsymbol{t} \times (\boldsymbol{R}\boldsymbol{x})] = 0 \quad \Rightarrow \quad \boldsymbol{x}'^T [\boldsymbol{t}_\times] \boldsymbol{R}\boldsymbol{x} = 0$$

The vectors ***Rx***, ***t***, and ***x'*** are coplanar

Recall: $\mathbf{a} \times \mathbf{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [\mathbf{a}_\times]\mathbf{b}$

$[a_x]$ is the skew symmetric matrix of a

Epipolar constraint: Calibrated case



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

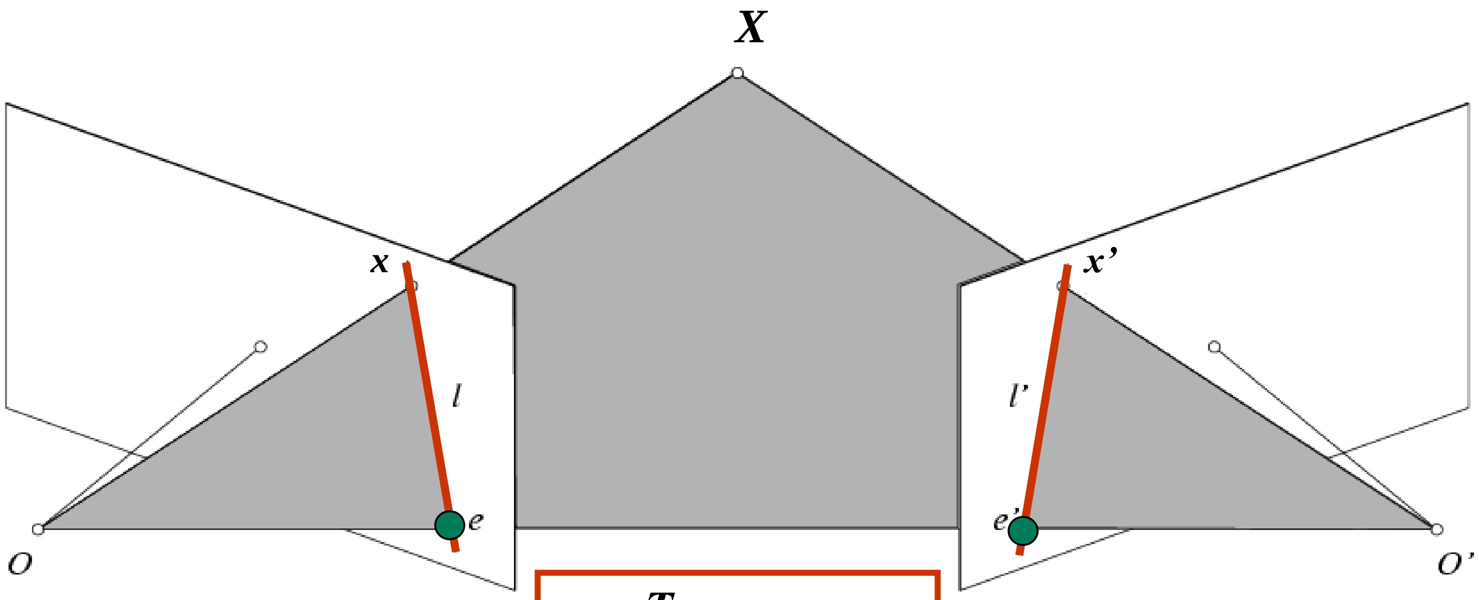
The vectors $\mathbf{Rx}$, $\mathbf{t}$, and $\mathbf{x'}$ are coplanar

E is a 3x3 matrix, which relates corresponding pairs of normalized homogeneous image points across pairs of images – for calibrated cameras.

Essential Matrix
(Longuet-Higgins, 1981)

Estimates relative position/orientation (camera pose)

Epipolar constraint: Calibrated case



$$\boxed{x'^{T} E x = 0}$$

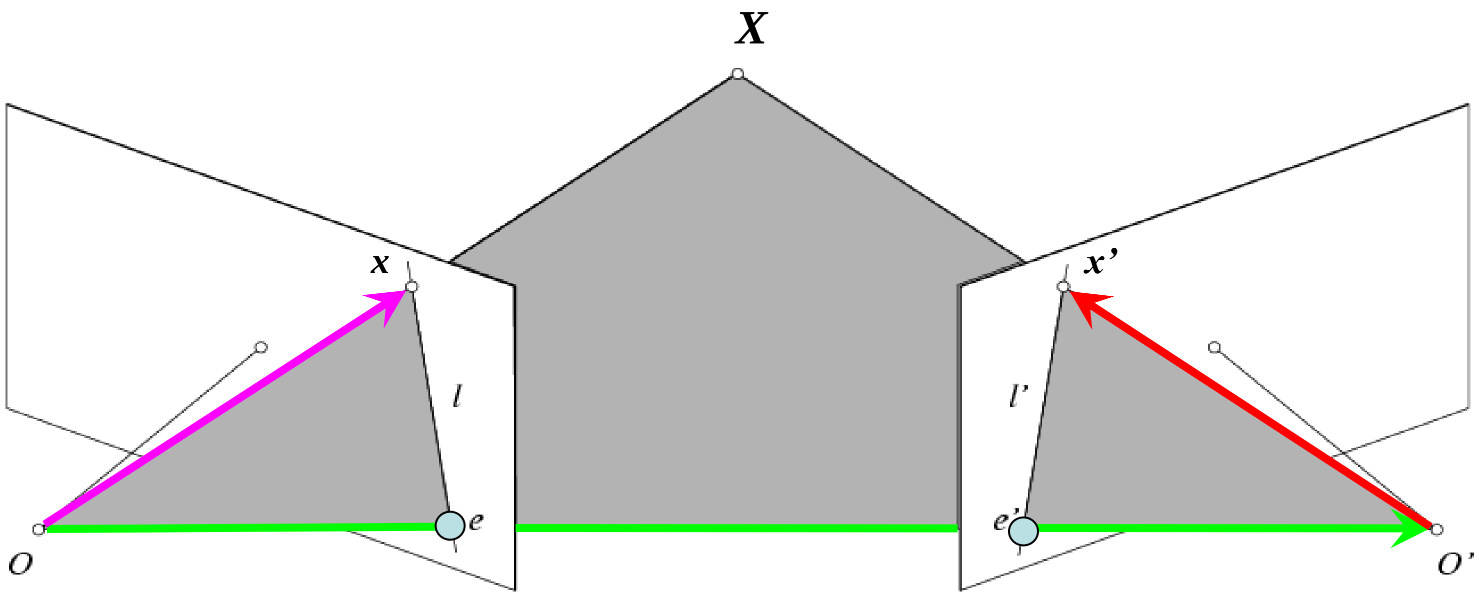
Essential Matrix

- ***E x*** is the epipolar line associated with ***x*** (***l'*** = ***E x***)
  - Recall: a line is given by $ax + by + c = 0$ or

$$\mathbf{l}^T\mathbf{x} = 0 \quad \text{where} \quad \mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- ***E x*** is the epipolar line associated with ***x*** (***l'*** = ***E x***)
- $\boldsymbol{E}^T\boldsymbol{x}'$ is the epipolar line associated with ***x'*** ($\boldsymbol{l} = \boldsymbol{E}^T\boldsymbol{x}'$)
- $\boldsymbol{E}\,\boldsymbol{e} = 0$ and $\boldsymbol{E}^T\boldsymbol{e}' = 0$
- ***E*** is singular (rank two)
- ***E*** has five degrees of freedom (3 for R, 2 for t because it's *up to a scale*)

Epipolar constraint: Uncalibrated case



- The calibration matrices ***K*** and ***K'*** of the two cameras are **unknown**
- We can write the epipolar constraint in terms of *unknown* normalized coordinates (**pixels**):

$$\hat{x}'^{T} E \, \hat{x} = 0 \qquad \hat{x} = K^{-1} x, \quad \hat{x}' = K'^{-1} x'$$

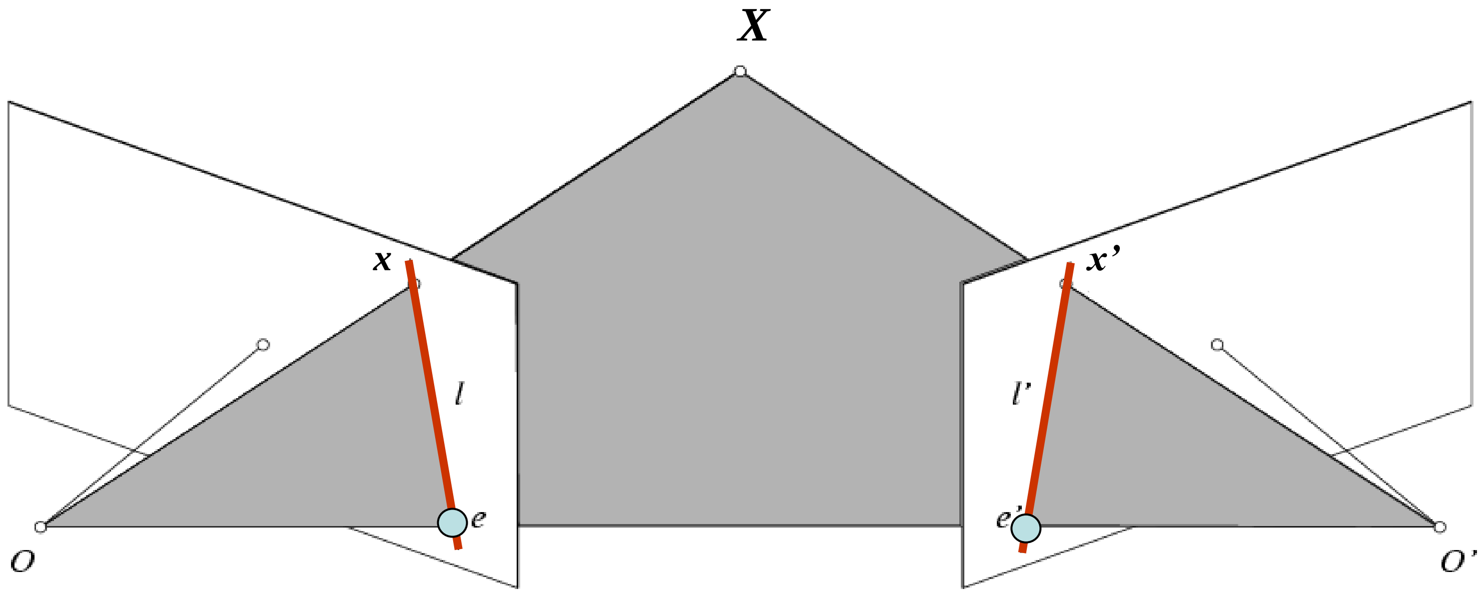
$$x_{\text{norm}} = K^{-1} x_{\text{pixel}} \quad x'_{\text{norm}} = K'^{-1} x'_{\text{pixel}}$$

$$\hat{x}'^{T} E \, \hat{x} = 0 \Longrightarrow x'^{T} F \, x = 0 \quad \text{with} \quad F = K'^{-T} E \, K^{-1}$$

Fundamental Matrix
(Faugeras and Luong, 1992)

It depends on ***intrinsic*** and ***extrinsic*** parameters

Epipolar constraint: Uncalibrated case



$$\hat{\mathbf{x}}'^T \boldsymbol{E}\, \hat{\mathbf{x}} = 0 \quad \Rightarrow \quad \boldsymbol{x}'^T \boldsymbol{F}\, \boldsymbol{x} = 0 \quad \text{with} \quad \boldsymbol{F} = \boldsymbol{K}'^{-T} \boldsymbol{E}\, \boldsymbol{K}^{-1}$$

Fundamental Matrix

- $\boldsymbol{F}\,\boldsymbol{x}$ is the epipolar line associated with $\boldsymbol{x}$ ($\boldsymbol{l}' = \boldsymbol{F}\,\boldsymbol{x}$)
- $\boldsymbol{F}^T\boldsymbol{x}'$ is the epipolar line associated with $\boldsymbol{x}'$ ($\boldsymbol{l} = \boldsymbol{F}^T\boldsymbol{x}'$)
- $\boldsymbol{F}\,\boldsymbol{e} = 0$ and $\boldsymbol{F}^T\boldsymbol{e}' = 0$
- ***F*** is singular (rank two): det(F)=0
- ***F*** has *eight* degrees of freedom (9 entries but defined up to scale)
- **It is not possible to recover the camera pose from F**, since it is composed of K (5 dof), R (3 dof) and t (3-1 dof), thus 10 dof

Example

$$\boldsymbol{Fx} \quad \begin{pmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{pmatrix} \begin{pmatrix} 343.53 \\ 221.70 \\ 1.0 \end{pmatrix}$$

x = 343.5300 y = 221.7005

$$\begin{matrix} 0.0295 \\ 0.9996 \\ -265.1531 \end{matrix}$$

$\boldsymbol{F\,x}$ is the **epipolar line** associated with $\boldsymbol{x}$ ($\boldsymbol{l'} = \boldsymbol{F\,x}$)

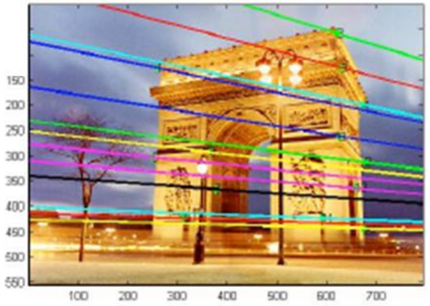
Where is the epipole?

$F * e_L = 0$

vector in the right nullspace of matrix F.

However, due to noise, F may not be singular.

So instead, next best thing is **eigenvector** associated with **smallest eigenvalue** of F




```
>> [u,d] = eigs(F' * F)
```

u =

$$\begin{matrix} -0.0013 & 0.2586 & \boxed{-0.9660} \\ 0.0029 & -0.9660 & \boxed{-0.2586} \\ 1.0000 & 0.0032 & \boxed{-0.0005} \end{matrix}$$

d = 1.0e8*

$$\begin{matrix} -1.0000 & 0 & 0 \\ 0 & -0.0000 & 0 \\ 0 & 0 & -0.0000 \end{matrix}$$

eigenvector associated with smallest eigenvalue

```
>> uu = u(:,3)
```

uu = (-0.9660 -0.2586 -0.0005)

```
>> uu / uu(3)
```

: to get pixel coords

(1861.02 498.21 1.0)

Estimating the Fundamental Matrix: 8-point algorithm

- Assume that you have *m* correspondences
- Each correspondence satisfies:

$$\bar{p_r}_i^T F \bar{p_l}_i = 0 \quad i = 1, \ldots, m$$

- F is a 3x3 matrix (9 entries)
- Set up a homogenous linear system with 9 unknowns, Af=0
- For estimating the essential/fundamental matrix, each point only contributes one constraint (row). [*because the Longuet-Higgins / Epipolar constraint is a scalar equation*]
- Thus need at least 8 points.

- **8-point algorithm**:
  1. Least squares solution using SVD on equations from 8 (*or more*) pairs of correspondences
  2. Enforce det(F)=0 constraint using SVD on F, since F must be singular (remember, it is rank 2)

- We can use the 8-point algorithm also to estimate the *Essential matrix E* (if we have calibrated cameras). *For E we can use also a non-linear 5-point algorithm.*

Note: estimation of F (or E) is degenerate for a planar scene.

8-point algorithm

1. Solve a system of homogeneous linear equations
   a. Write down the system of equations

$$\bar{p}_{l\,i} = (x_i \;\; y_i \;\; 1)^T \quad \bar{p}_{r\,i} = (x'_i \;\; y'_i \;\; 1)^T$$

One point

$$\bar{p}_{r\,i}^{\,T} F \bar{p}_{l\,i} = 0 \quad i = 1, \ldots, m$$

$$\begin{bmatrix} x'_i & y'_i & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

$$\begin{gathered} x_i x'_i f_{11} + x_i y'_i f_{21} + x_i f_{31} + \\ y_i x'_i f_{12} + y_i y'_i f_{22} + y_i f_{32} + \\ x'_i f_{13} + y'_i f_{23} + f_{33} = 0 \end{gathered}$$

One equation

8-point algorithm

1. Solve a system of homogeneous linear equations
   a. Write down the system of equations

Given m point correspondences…

$$\begin{bmatrix} x_1x'_1 & x_1y'_1 & x_1 & y_1x'_1 & y_1y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx'_m & x_my'_m & x_m & y_mx'_m & y_my'_m & y_m & x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0$$

We want the **eigenvector with smallest eigenvalue**

We can find the eigenvectors and eigenvalues of A^TA by finding the Singular Value Decomposition of A

Matlab:

```
[U, S, V] = svd(A);
f = V(:, end);
F = reshape(f, [3 3])';
```

1. Resolve det(F) = 0 constraint by using SVD

Matlab:

```
[U, S, V] = svd(F);
S(3,3) = 0;
F = U*S*V';
```

F must be singular (remember, it is rank 2, since it is important for it to have a left and right nullspace, i.e. the epipoles).

To enforce rank 2 constraint:

- Find the SVD of F: $F = U_f D_f V_f^T$
- Set smallest s.v. of F to 0 to create D'_f
- Recompute F: $F = U_f D'_f V_f^T$

From epipolar geometry to camera pose

- Estimating the fundamental matrix is known as "**weak calibration**"
- If we know the **calibration matrices** of the two cameras, we can estimate the essential matrix: $E = K'^{T}FK$
- The essential matrix gives us the relative rotation and translation between the cameras, or their **extrinsic parameters**

The Fundamental Matrix Song: http://danielwedge.com/fmatrix/

Università di Genova | DIBRIS DIPARTIMENTO DI INFORMATICA, BIOINGEGNERIA, ROBOTICA E INGEGNERIA DEI SISTEMI

3D reconstruction

3D reconstruction: Stereo Reconstruction

- Given point correspondences, how to compute 3D point positions using triangulation.

- Results depend on how calibrated the system is:

1. Intrinsic and extrinsic parameters known

   Can compute metric 3D geometry

2. Only intrinsic parameters known (and E)

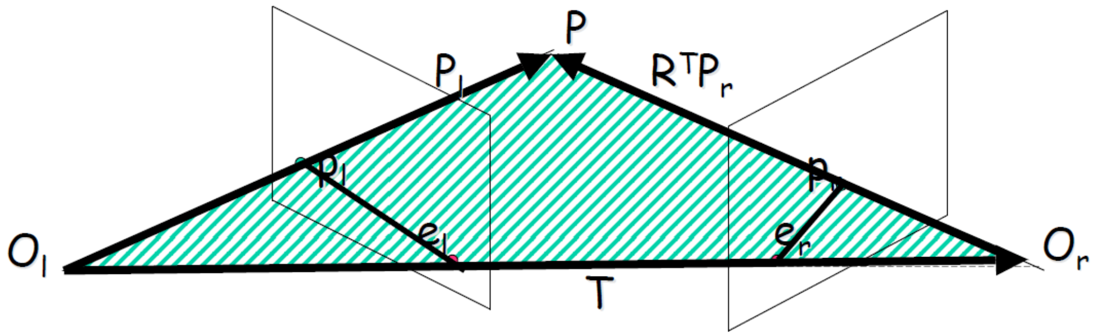
   Can compute 3D geometry up an unknown scale factor

3. *Neither intrinsic nor extrinsic known*

   Recover structure up to an unknown projective transformation of the scene

Fully Calibrated Stereo: Calibrated Triangulation

- Known intrinsics: can compute **viewing rays** in camera coordinate system
- Know extrinsics: know how **rays** from both cameras are **positioned** in **3D** space
- Reconstruction: **triangulation** of viewing rays



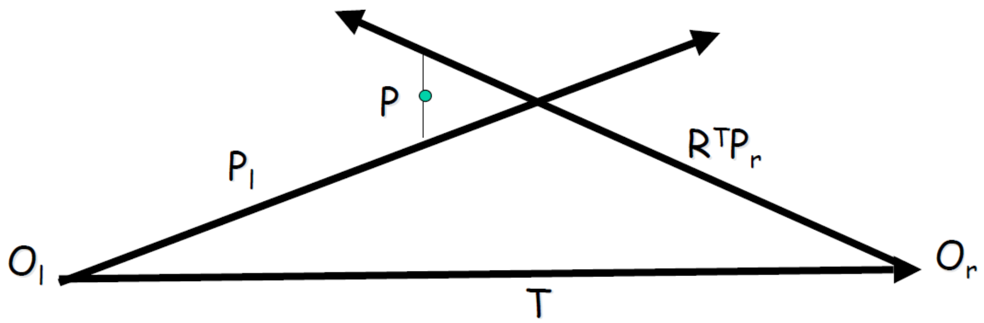
ideally, P is the point of intersection of two 3D rays:

ray through O_l with direction P_l

ray through O_r with direction R^TP_r

Fully Calibrated Stereo: Calibrated Triangulation

- Unfortunately, these rays typically *don't intersect* due to **noise** in point locations and calibration parameters



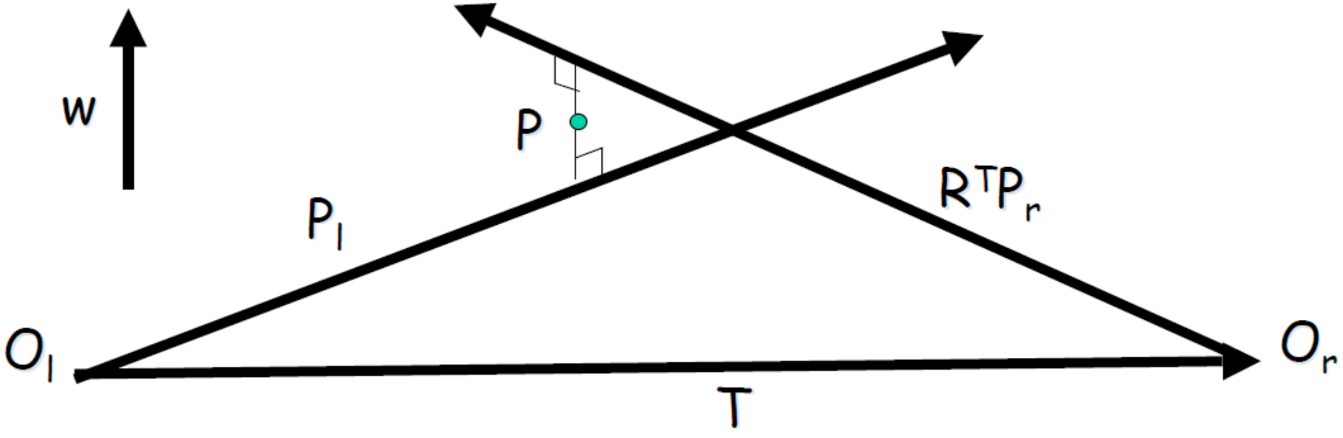
- **Solution**: Choose P as the "*pseudo-intersection point*". This is point that minimizes the sum of squared distance (SSD) to both rays. (The SSD is 0 if the rays exactly intersect)

Fully Calibrated Stereo: Calibrated Triangulation

A possible solution

P is midpoint of the segment perpendicular to P_l and $R^T P_r$

Let $w = P_l \times R^T P_r$ (this is perpendicular to both)

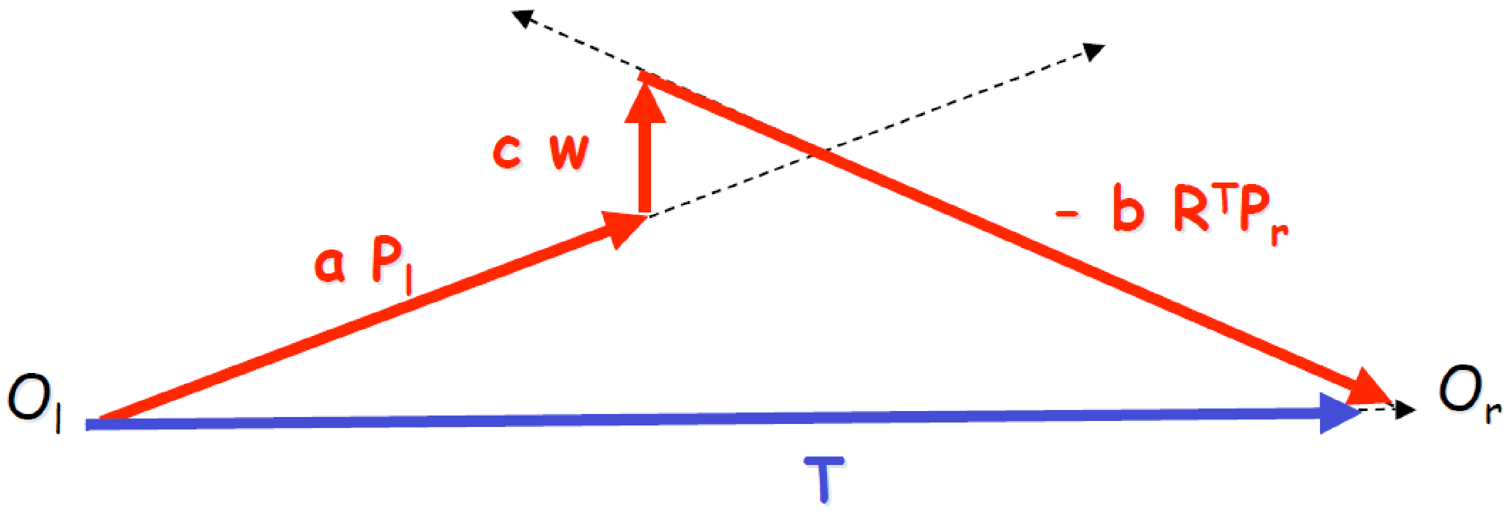


Introducing three unknown scale factors a,b,c we note we can write down the equation of a "circuit"

Fully Calibrated Stereo: Calibrated Triangulation

Writing vector "circuit diagram" with unknowns a,b,c

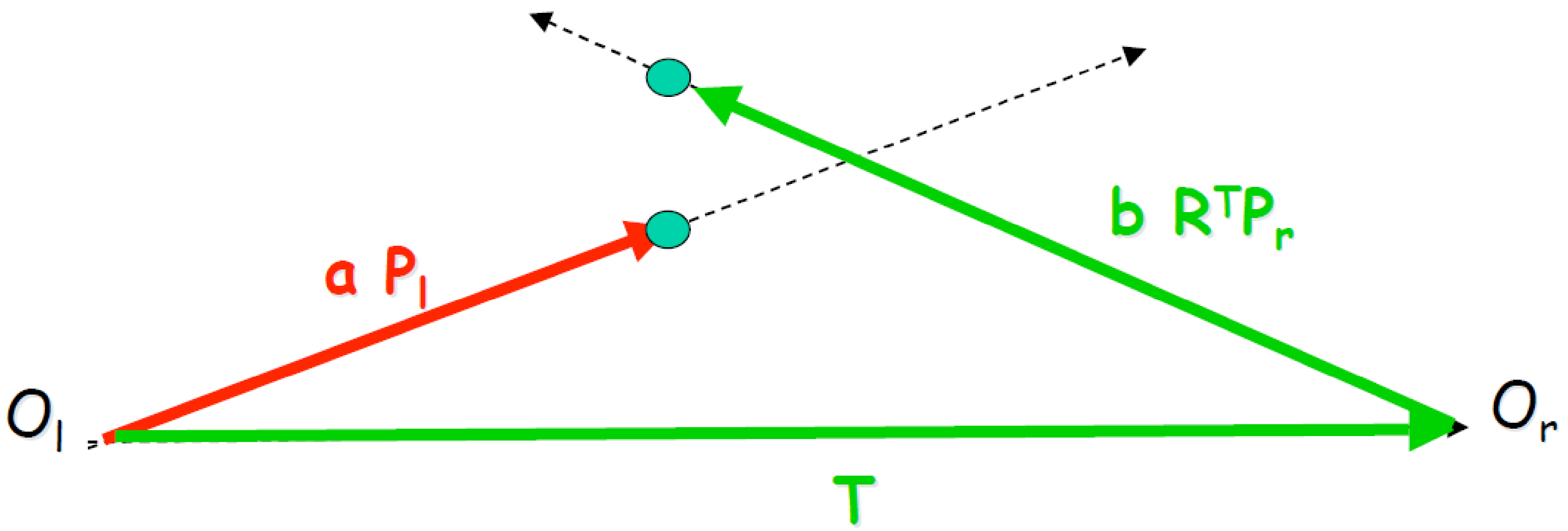
$$a\,P_l + c\,(P_l \times R^T P_r) - b\,R^T P_r = T$$



note: this is three linear equations in three unknowns a,b,c
=> can solve for a,b,c

Fully Calibrated Stereo: Calibrated Triangulation

After finding a,b,c , solve for midpoint of line segment between points $O_l + a\ P_l$ and $O_l + T + b\ R^T\ P_r$



Only Intrinsic Parameters Known: 3D reconstruction

- Use knowledge that $E = [t_x]R$ to solve for R and T, then use previous triangulation method.
- Note: since E is only defined up to a scale factor, we can only determine the **direction** of T, **not** its **length**.
- So... 3D reconstruction will have an **unknown scale**.

- But *this scale can be determined if we know the distance between two points in the observed scene or the distance between the two cameras.*

Only Intrinsic Parameters Known: 3D reconstruction

Using E to solve for extrinsic parameters R and T

- $E = R\ S$ where elements of S are functions of T
- Then $E^T E = S^T R^T R S = S^T S$ (because $R^T R = I$)
- Thus, $E^T E$ is only a function of T.
- Solve for elements of T *by assuming it is a unit vector*.
- After determining T, plug back into $E = R\ S$ to determine R.

Only Intrinsic Parameters Known: 3D reconstruction

- Unfortunately, four different solutions for (R,T) are possible (due to the choice of sign of E, and choice of sign of T when solving for it).

- However, only one choice will give *consistent solutions* when used for triangulation, where consistent means reconstructed points are in front of the cameras (positive Z coordinates).

- So, check all four solutions, choose the correct one, and you are done.

3D reconstruction: intrinsics known

- We introduce **another algorithm** that uses the relative pose (rotation and translation) between the two cameras (*up to an arbitrary scale*) by considering Essential matrix.

- Relative pose and point correspondences can then be used to retrieve the position of the points in 3-D by recovering *their depths relative to each camera frame*.
- Consider the basic rigid-body equation, where the pose (R, T) has been recovered, in terms of the **images** and the **depths**, it is given by

$\lambda x = RX + t$

$$\lambda_2^j x_2^j = \lambda_1^j R x_1^j + \gamma T, \quad j = 1, 2, \ldots, n.$$

- Notice that since (R, T) are known, the equations are linear in both the depth λ's and the ***scale*** γ.

3D reconstruction: intrinsics known

- For each point, λ_1, λ_2 are its depths with respect to the first and second camera frames, respectively. One of them is redundant, it is simply a function of (R, T).
- Hence, we can eliminate, say, λ_2 from the above equation by multiplying both sides by $\widehat{x_2}$ (cross product as matrix), which yields

$$\lambda_1^j \widehat{x_2^j} R x_1^j + \gamma \widehat{x_2^j} T = 0, \quad j = 1, 2, \ldots, n.$$

- This is equivalent to solve

$$M^j \bar{\lambda}^j \doteq \left[\widehat{x_2^j} R x_1^j, \ \widehat{x_2^j} T \right] \begin{bmatrix} \lambda_1^j \\ \gamma \end{bmatrix} = 0,$$

for all *n* equations.

3D reconstruction: intrinsics known

- Since they share the same scale γ, we define

$$\vec{\lambda} = [\lambda_1^1, \lambda_1^2, \ldots, \lambda_1^n, \gamma]^T \in \mathbb{R}^{n+1} \qquad M \in \mathbb{R}^{3n \times (n+1)}$$

$$M \doteq \begin{bmatrix} \widehat{x_2^1} R x_1^1 & 0 & 0 & 0 & 0 & \widehat{x_2^1} T \\ 0 & \widehat{x_2^2} R x_1^2 & 0 & 0 & 0 & \widehat{x_2^2} T \\ 0 & 0 & \ddots & 0 & 0 & \vdots \\ 0 & 0 & 0 & \widehat{x_2^{n-1}} R x_1^{n-1} & 0 & \widehat{x_2^{n-1}} T \\ 0 & 0 & 0 & 0 & \widehat{x_2^n} R x_1^n & \widehat{x_2^n} T \end{bmatrix}$$

- Then the equation

$$M\vec{\lambda} = 0$$

determines all the unknown depths *up to a single universal scale.*

3D reconstruction: intrinsics known

- The linear least squares estimate of $\vec{\lambda}$ is simply the eigenvector of M^TM that corresponds to its smallest eigenvalue.

- Note that this scale ambiguity is intrinsic, since without any prior knowledge about the scene and camera motion, one cannot disambiguate whether the camera moved twice the distance while looking at a scene twice larger but two times further away.

  <u>This scale can be determined if we know the distance between two points in the observed scene or the distance between the two cameras</u>.

Optimal pose and structure: bundle adjustment

- Use 8-point algorithm to get initial value of F (or E)
- Jointly solve for 3D points **X** and **F (**or **E)** that minimize the squared <u>*reprojection error*</u>

- **Bundle adjustment**:

  in practice, we cannot measure the actual coordinates but only their noisy versions,

$$\tilde{x}_1^j = x_1^j + w_1^j, \quad \tilde{x}_2^j = x_2^j + w_2^j, \quad j = 1, 2, \ldots, n$$

One can write the optimization problem in unconstrained form:

$$\sum_{j=1}^{n} \left\|\tilde{x}_1^j - \pi_1(X^j)\right\|_2^2 + \left\|\tilde{x}_2^j - \pi_2(X^j)\right\|_2^2,$$

where π_1 and π_2 denote the projection of a point ***X*** in space onto the first and second images, respectively.

If we choose the first camera frame as the reference, then the above expression can be simplified to

$$\phi(x_1, R, T, \lambda) = \sum_{j=1}^{n} \left\|\tilde{x}_1^j - x_1^j\right\|_2^2 + \left\|\tilde{x}_2^j - \pi(R\lambda_1^j x_1^j + T)\right\|_2^2.$$

Minimizing the above expression with respect to the unknowns
(R; T; x_1; λ) is named in the literature *bundle adjustment*.

The minimization is performed by using nonlinear least-squares algorithms, such as Levenberg–Marquardt.

RGB-D cameras and stereo displays

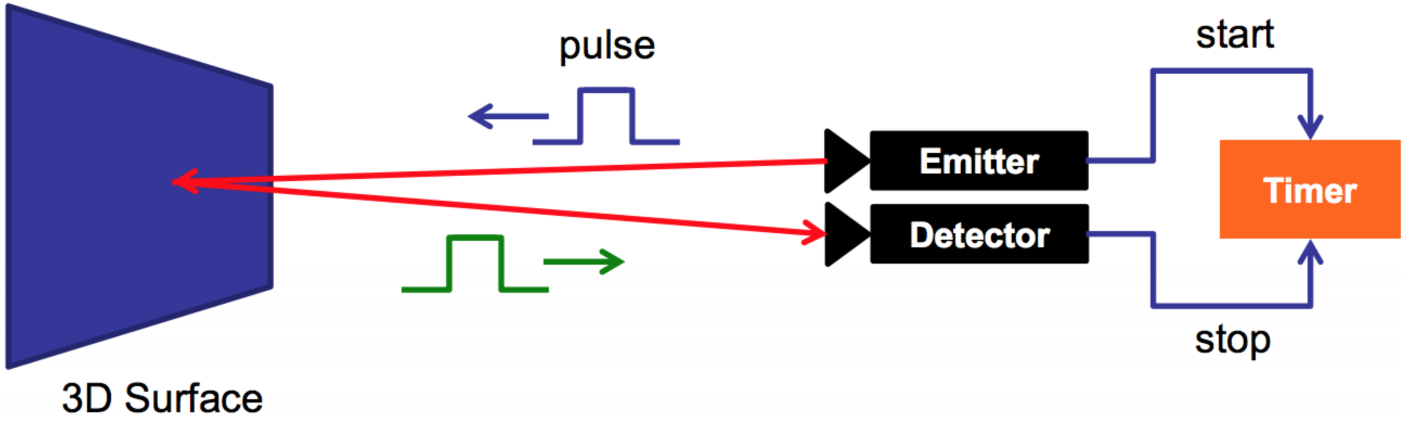
How does a depth camera work?

- Passive illumination (standard RGB cameras)
  - Natural (or existing) light sources
  - Visible spectrum 380-780nm
  - **Visual features** (e.g. SSD, corners)
  - Cannot track when it is too dark (mostly indoors)

- Active illumination (RGB-D cameras)
  - Often **infrared spectrum**
  - LED beacons
  - Camera with infrared filter delivers high contrast
  - Not suitable with sunlight

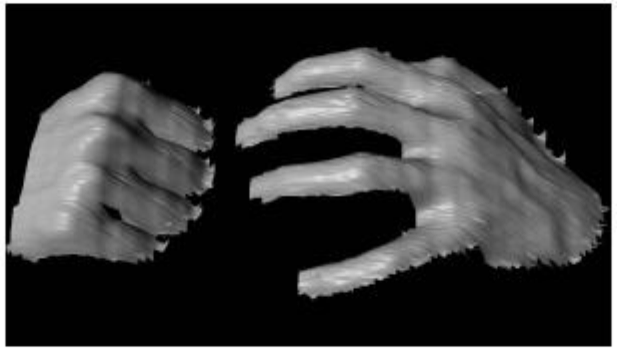
Active stereo: time of flight

- Depth cameras in HoloLens (and Kinect V2) use *time of flight*
  - "sonar for light"
  - Emit light of a known wavelength, and time how long it takes for it to come back

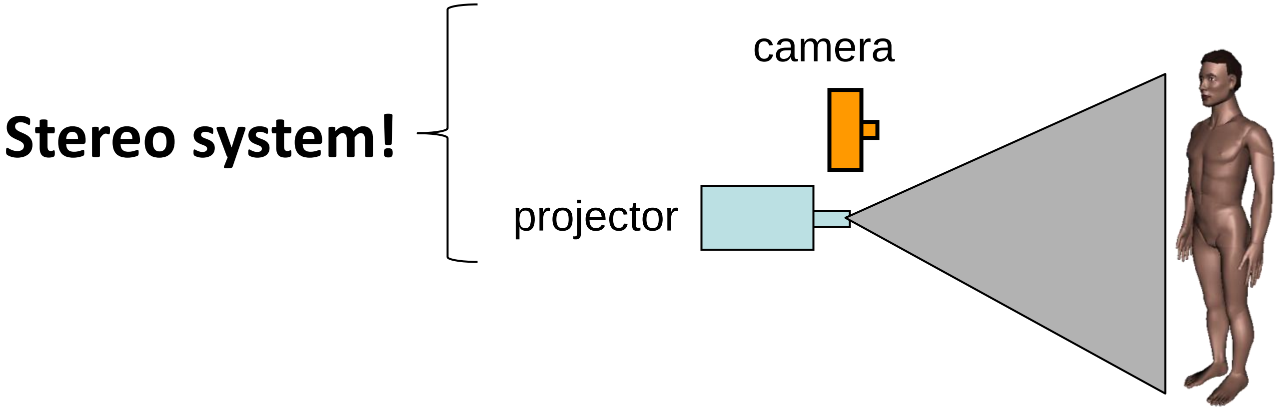


Active stereo: structured light





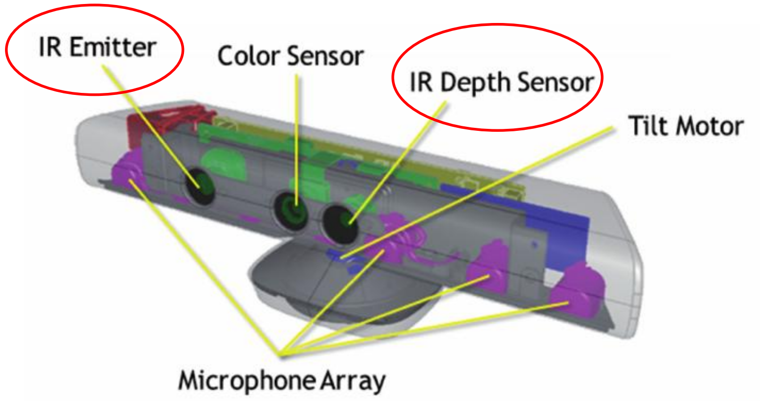
- Camera projects "structured" light patterns onto the object
  - Simplifies the correspondence problem
  - Allows us to use only one camera



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

Active stereo: structured light

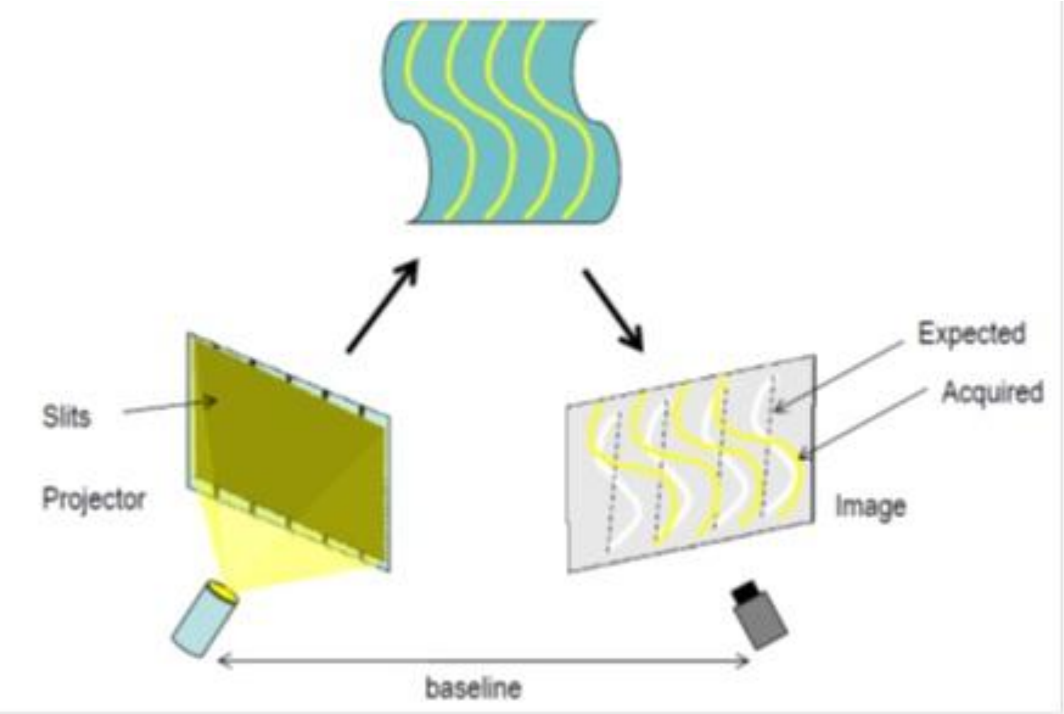
- The depth map is constructed by analyzing a speckle pattern of infrared light (*structured light*)
- Structured light general principle: *to project a known pattern onto the scene and to infer depth from the deformation of that pattern*



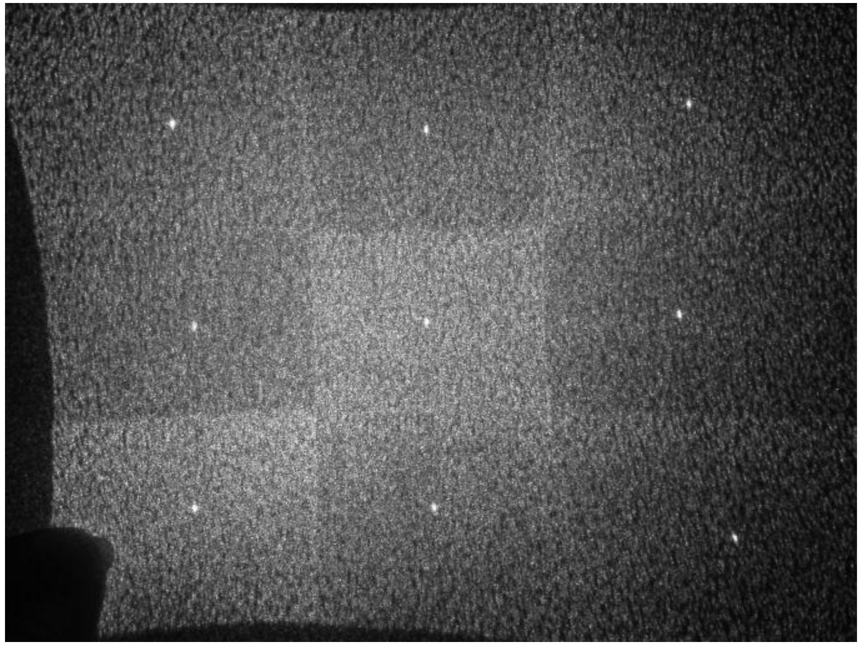
A. Canessa, Andrea, M. Chessa, A. Gibaldi, S.P. Sabatini, and F. Solari. "*Calibrated depth and color cameras for accurate 3D interaction in a stereoscopic augmented reality environment*." Journal of Visual Communication and Image Representation 25, no. 1, pp: 227-237, 2014.

Active stereo: structured light

Structured light principle

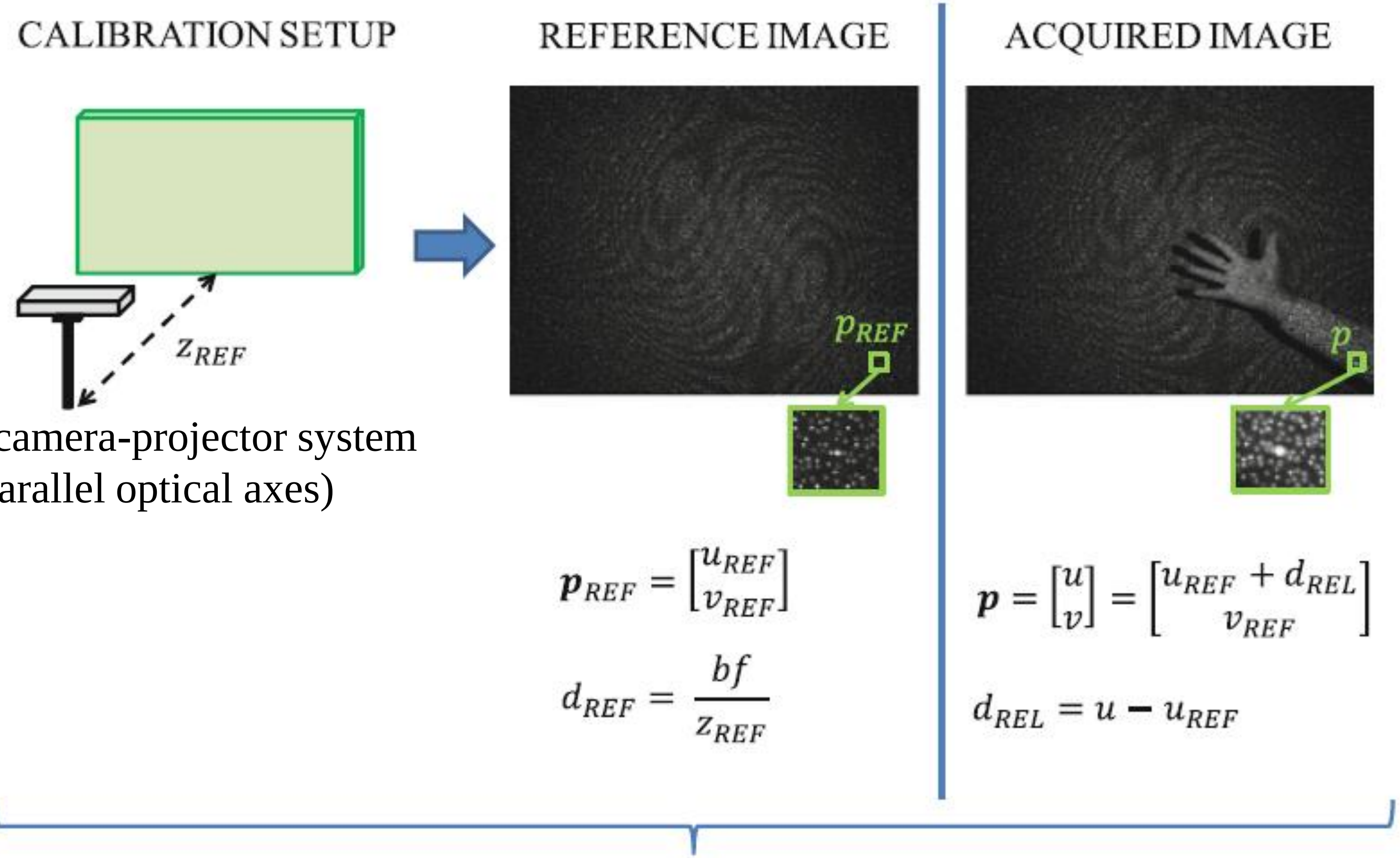


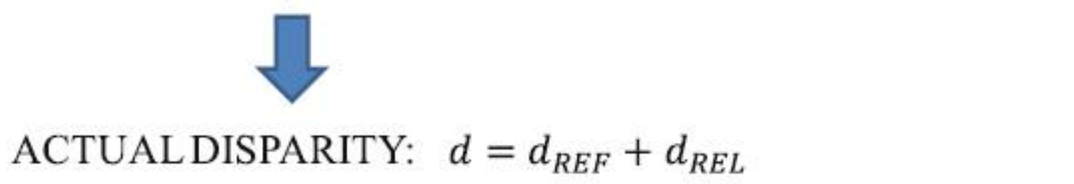
Speckle pattern



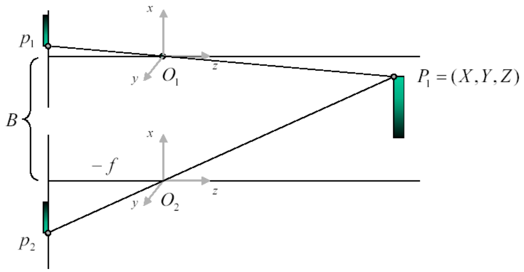
The objective of structured light systems is **to simplify the correspondence problem** through projecting effective patterns by the illuminator: **to infer depth from the deformation of that pattern**.

Structured light: camera virtualization





stereo camera-projector system
(parallel optical axes)

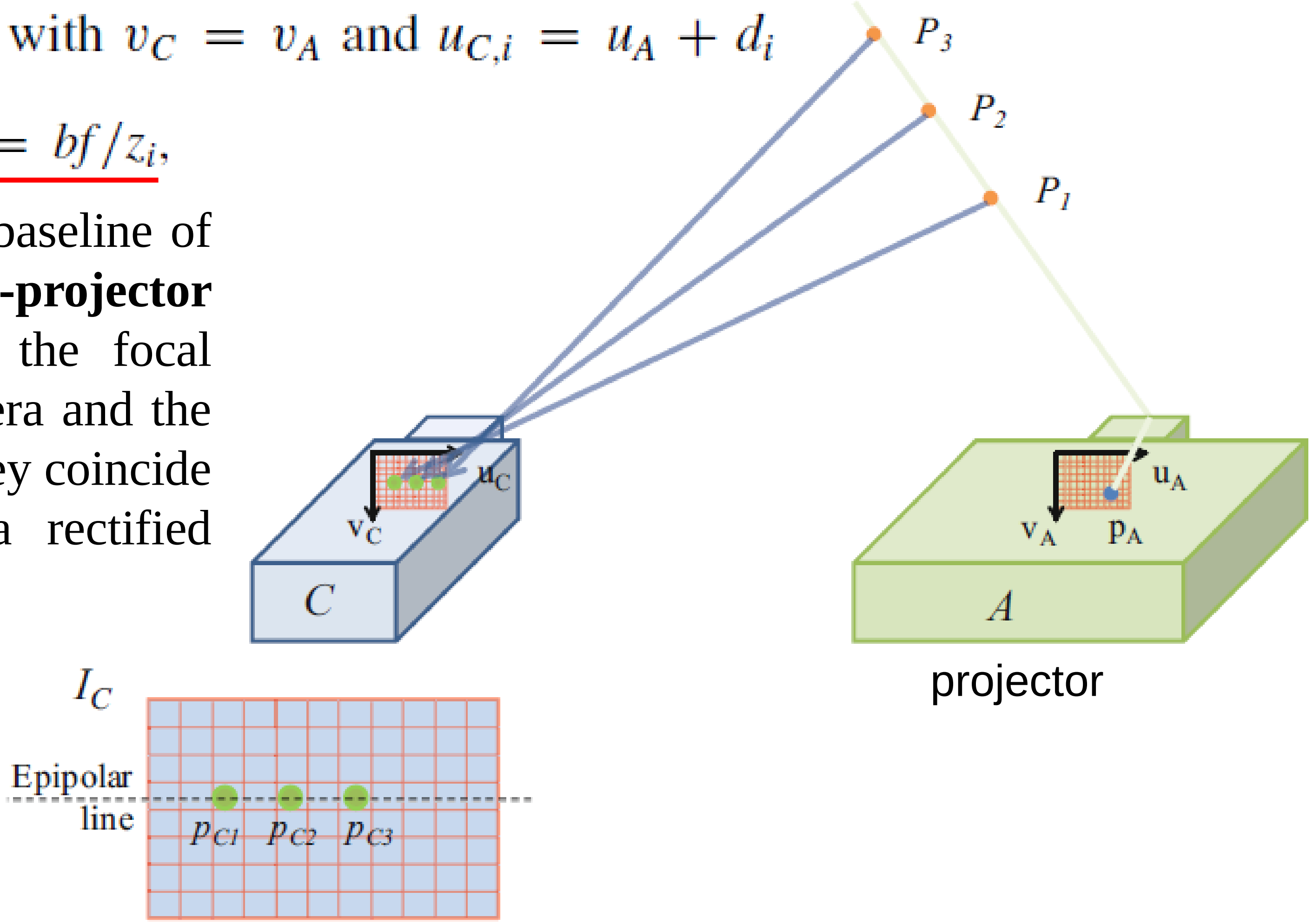


Structured light: camera virtualization

$$\mathbf{p}_{C,i} = [u_{C,i}, v_C] \text{ with } v_C = v_A \text{ and } u_{C,i} = u_A + d_i$$

with disparity $d_i = bf/z_i$,

In which *b* is the baseline of the **stereo camera-projector system** and *f* is the focal length of the camera and the projector (since they coincide in the case of a rectified system)



Structured light: camera virtualization

The disparity of each pixel ***P**c,i* can be expressed as a disparity *difference* or *relative disparity* with respect to a selected disparity reference.

In particular, if the selected disparity reference is *dREF =d2*, the values of *d1* and *d3* can be expressed with respect to *d2* as signed difference *dREL1 = d1 - d2* and *dREL3 = d3 - d2*.

Given the value of *zREF = z2* (calibration) and of *dREL1* and *dREL3* (computed), the value of *z1* and of *z3* can be obtained as

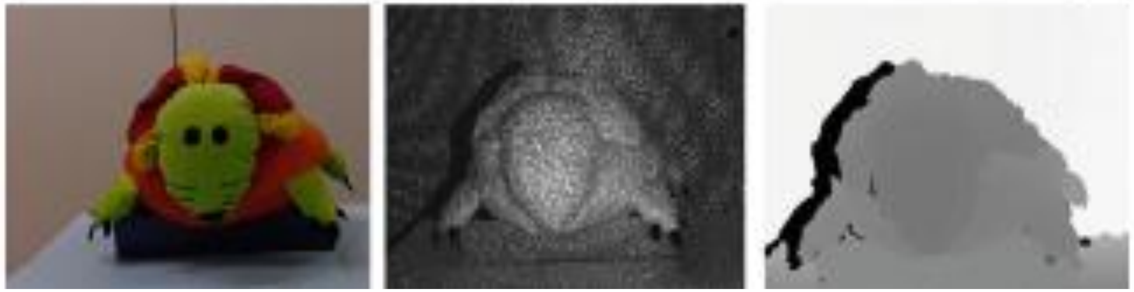
$$\Delta z_i = \frac{1}{\dfrac{1}{z_2} + \dfrac{d_{REL_i}}{bf}} - z_2$$

$$z_i = z_2 + \Delta z_i, \quad i = 1, 3.$$

Structured light: issues

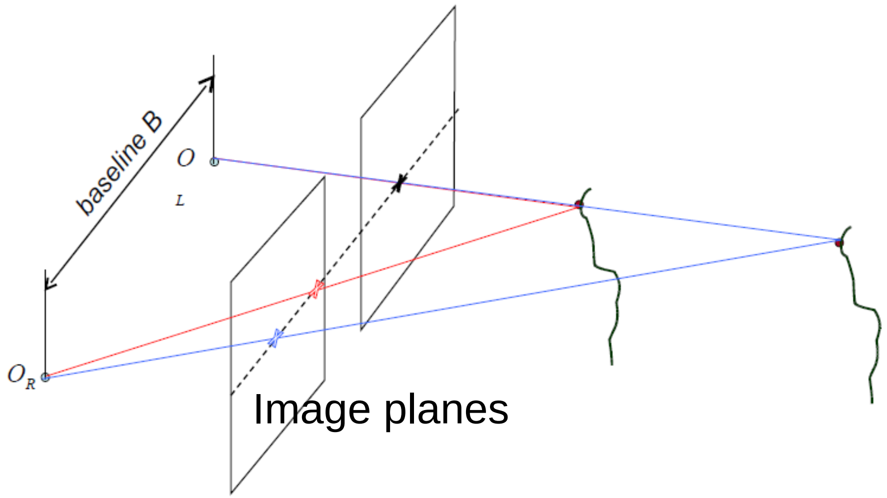
The dependence of the pattern appearance from the surface color.

A strong external illumination affects the acquired scene.



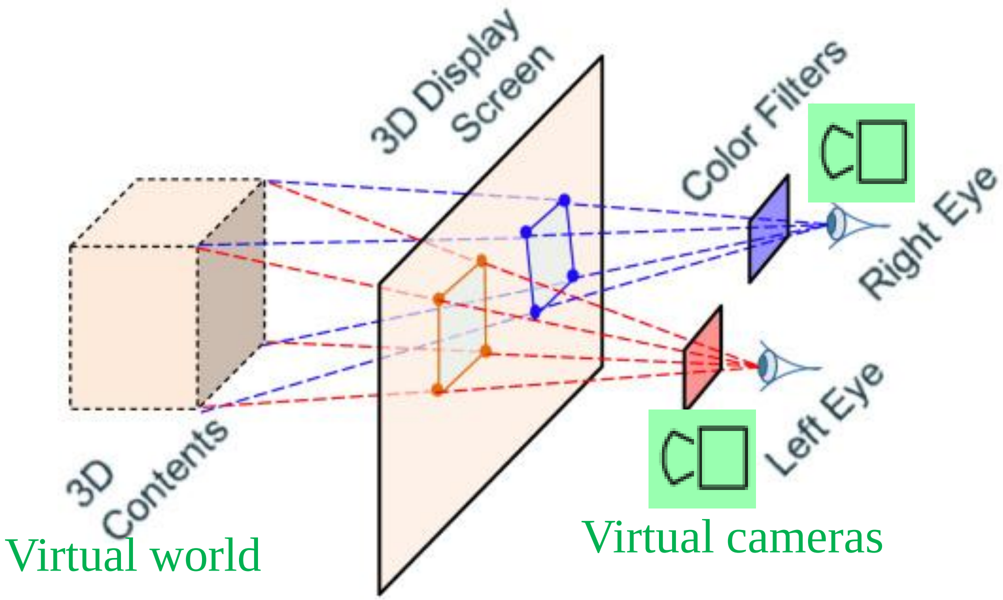
Occlusions: areas that are visible from the camera but not from the projector’s viewpoint

Stereoscopic vision

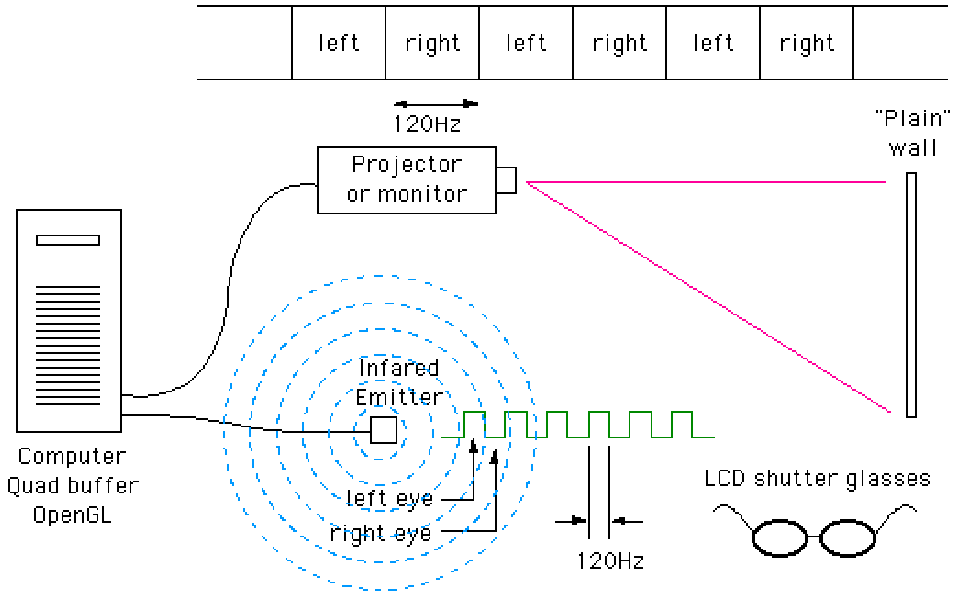


- Stereoscopic vision is based on information from 2 cameras (eyes) locations.

- Thus, to produce the sensation of depth (3D), the eyes must be elicited by two slightly different images (the stereo pairs).

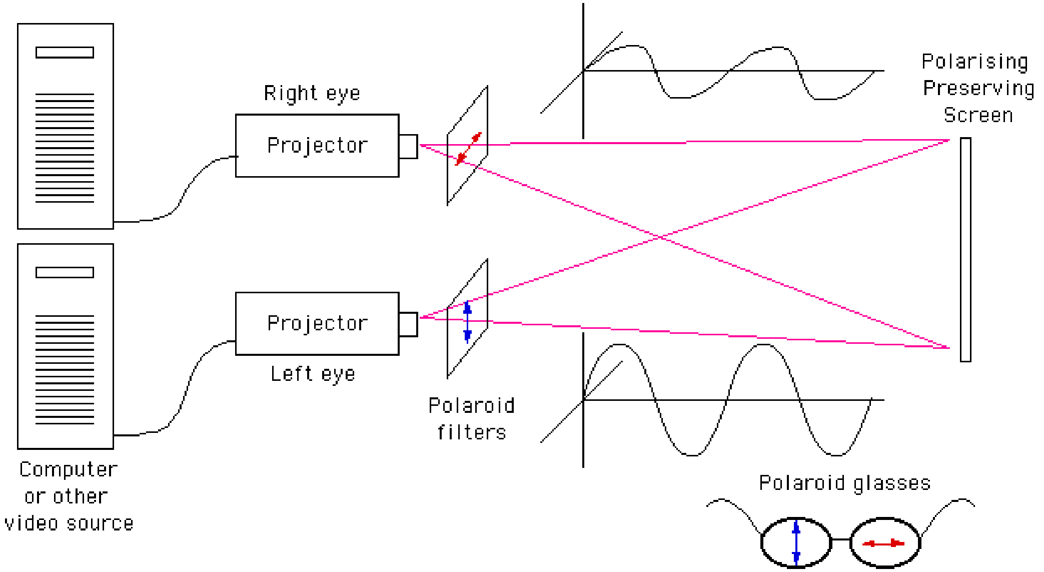


Steroscopic display



- Stereoscopic display: active technique

- Stereoscopic display: passive technique



Steroscopic HMD

How to create stereoscopic 3D images