Lecture 5
Epipolar Geometry

Professor Silvio Savarese
Computational Vision and Geometry Lab

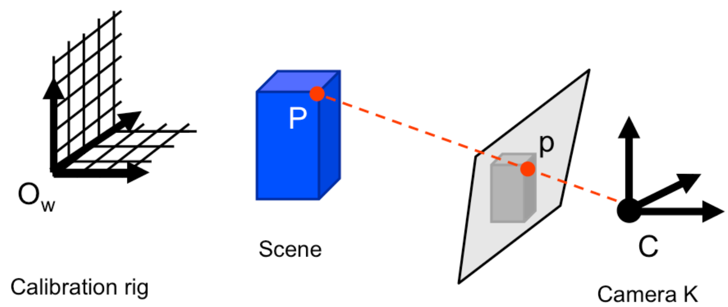
Lecture 5

Epipolar Geometry

- Why is stereo useful?
- Epipolar constraints
- Essential and fundamental matrix
- Estimating F
- Examples

Reading: [AZ] Chapter: 4 "Estimation – 2D perspective transformations
Chapter: 9 "Epipolar Geometry and the Fundamental Matrix Transformation"
Chapter: 11 "Computation of the Fundamental Matrix F"
[FP] Chapter: 7 "Stereopsis"
Chapter: 8 "Structure from Motion"

Recovering structure from a single view

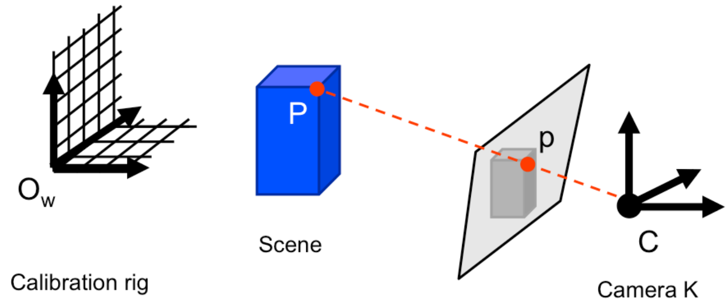


From calibration rig	→ location/pose of the rig, K
From points and lines at infinity + orthogonal lines and planes	→ structure of the scene, K
Knowledge about scene (point correspondences, geometry of lines & planes, etc…	

In the previous lectures, we have seen how to compute the intrinsic and extrinsic parameters of the camera using 1 or more view(s) – this is the camera calibration procedure.

We have also derived equations for estimating the intrinsics of the camera (i.e., the matrix K) from just one image (geometry of the vanishing points and lines) as well as for estimating some of the properties of the 3D world such as surface primitives, ground planes, etc…, given that some prior knowledge about the world is available (for instance, a building façade is orthogonal to the ground plane).

Recovering structure from a single view



Why is it so difficult?

Intrinsic ambiguity of the mapping from 3D to image (2D)

However, it is in general not possible to recover the structure of the 3D world (e.g., the blue box; the point P) from just one image. This is due to the intrinsic ambiguity of the 3D to 2D mapping. Any point along the ray from the camera center C to a point p in the image (line of sight) is a possible candidate for the actual 3D point P.

Recovering structure from a single view

Intrinsic ambiguity of the mapping from 3D to image (2D)

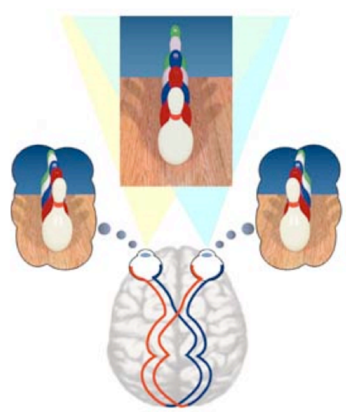
Courtesy slide S. Lazebnik

As we can see in this image, it is easy to fool our eyes. The pisa tower here is much further that the shoe of this person even if it looks like this person is kicking the tower!

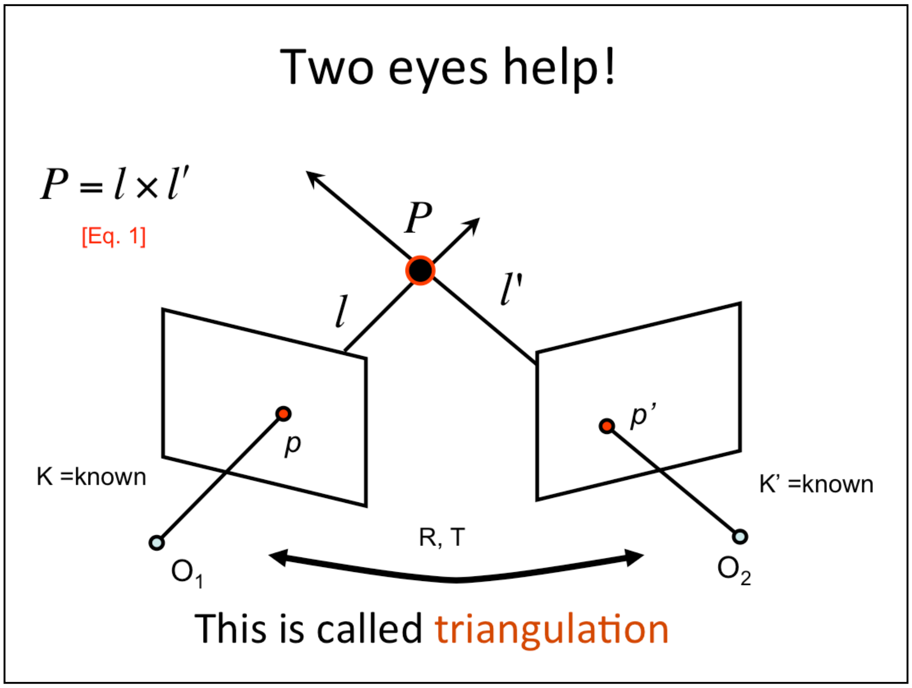
By careful inspection we could actually use the ground plane and the location of the object's base as cues, and can figure out that the tower is actually in the background and that the person is in the foreground.

If we use two eyes, or move our head a little bit, this illusion immediately disappears and we can perfectly figure out the correct scene layout.

Two eyes help!



So, what if there are two cameras observing the object? As we will shortly see, it turns out that it is quite helpful! Humans and most other life forms have evolved a pair of eyes so that the world around them can be perceived in 3D. Imagine how it would have been if we were only capable of perceiving 2D images!



Here's how having two eyes (or alternately, cameras) help. Suppose we have two cameras with known camera intrinsic parameters K and K', respectively, and known relative locations and orientations (i.e., we know the transformations R and T that relate the two camera reference systems). Suppose we have a point P in 3D. Let p and p' be the observations of P in the images of camera 1 and 2, respectively. The location of P is unknown in the 3D space, but we can measure the locations of p and p' in each image. Because K, K', R and T are known, we can compute the two lines of sight l and l' which are defined by O_1 and p, and O_2 and p', respectively. Thus, P can be computed as the intersection point between l and l' by Eq. 1. How do we compute this point if intersection? Through the cross product as illustrated in Eq. 1.

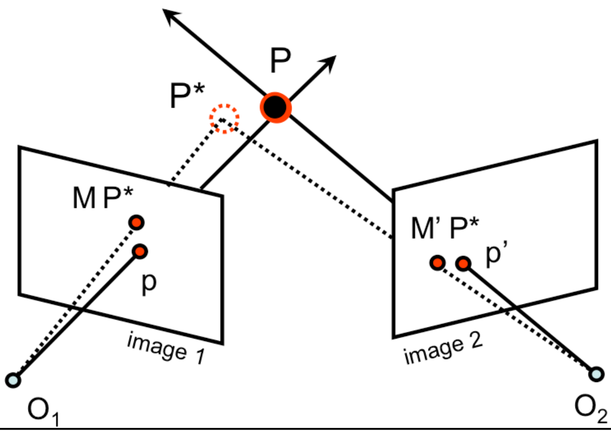
This process is known as Triangulation.

Does this work well in practice? No…

Triangulation

- Find P’ that minimizes

$$d(p, M\ P^*) + d(p', M' P^*) \quad \text{[Eq. 2]}$$



In practice, because the observations p and p’ are noisy and the camera calibration parameters may not be accurately estimated, finding the intersection point P using Eq.1 may be problematic (in fact, the exact intersection point may not exist at all). Thus, the triangulation problem is often mathematically characterized by solving the minimization problem expressed in Eq.2. In this equation, we seek to find a point P* in 3D that best approximates P by minimizing the re-projection error of P’ in both images. What is the re-projection error? The re-projection error for image 1 is computed as the distance between the re-projection of P* to the image 1 (via projective transformation M) and the corresponding observation p – that is, d(p, M P*). The re-projection error for image 2 is computed as the distance between the re-projection of P* to the image 2 (via projective transformation M’) and the corresponding observation p’ – that is, d(p’, M’ P*). The overall re-projection error is the sum of these two errors (Eq.2).

Stereo-view geometry

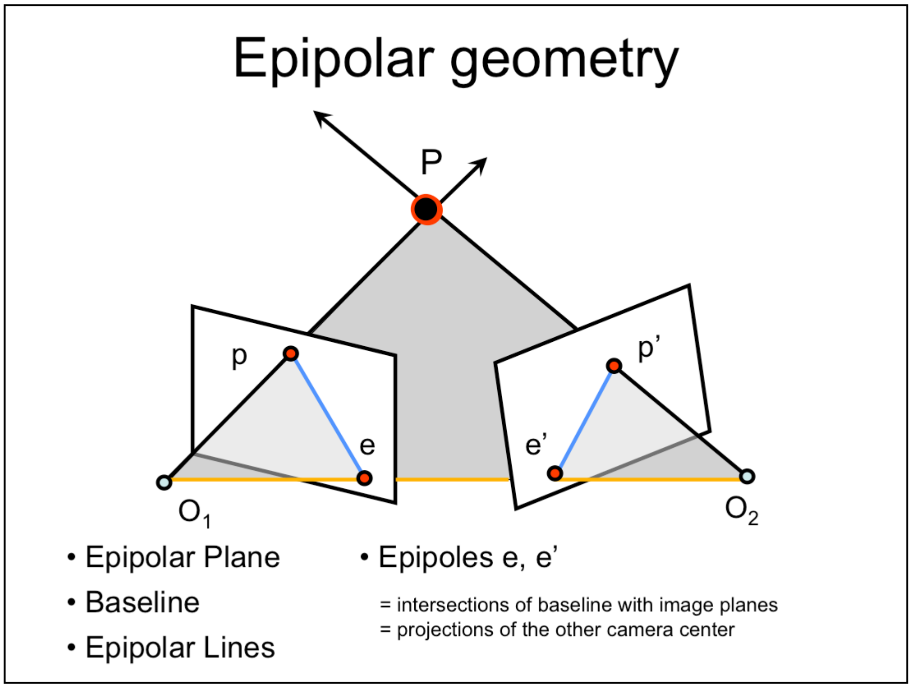
- **Correspondence:** Given a point p in one image, how can I find the corresponding point p’ in another one?
- **Camera geometry:** Given corresponding points in two images, find camera matrices, position and pose.
- **Scene geometry:** Find coordinates of 3D point from its projection into 2 or multiple images.

The key problems in multi-view (or stereo) geometry are:

i) inferring the camera geometry: Given corresponding points in two images, find camera matrices, position and pose.

ii) inferring the scene geometry, Find coordinates of 3D point from its projection into 2 or multiple images.

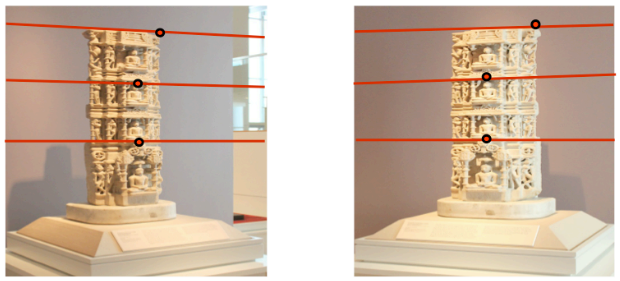
A key ingredient for doing so is to solve the correspondence problem: given the observations p and p’ in camera 1 and 2 respectively, how do we know that such observations correspond to the same 3D point P, or equivalently, that p and p’ are in correspondence? We will discuss this problem in more details in the next lecture. In this lecture will focus on the geometry that relates camera, points in 3D and corresponding observations– that is, we will focus on what we call the epipolar geometry of a stereo pair.



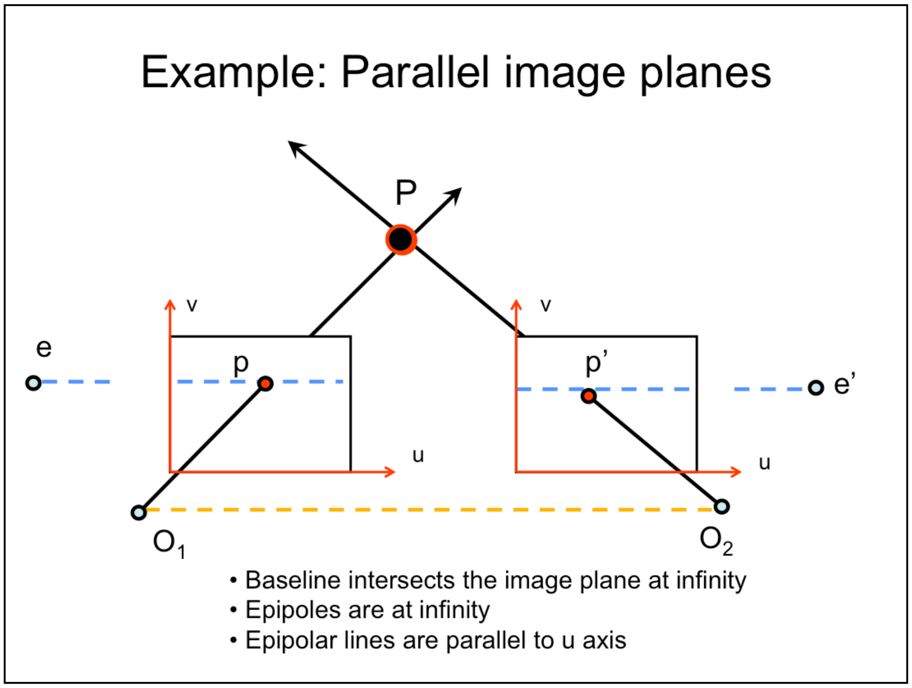
We introduce the following definitions. Let O_1 and O_2 be two camera centers. The line that connects O_1 and O_2 is called *baseline*. O_1 and O_2 and P define the *epipolar plane*. Let p and p’ be two corresponding points in images 1 and 2. The *epipoles* e and e’ are defined as the intersections of the baseline and the image planes of camera 1 and 2, respectively. The rays O_1 p and O_2 p’ intersect in the desired 3D coordinate P.

The intersection of the epipolar plane (associated to O_1 (or O_2) and the observations p and p’) with the camera images define *the epipolar lines*. The line defined by p and e is an example *epipolar line* in the image 1; The line defined by p’ and e’ is an example of *epipolar line* in the image 2. The epipolar lines have the property that they all intersect at the corresponding epipoles.

Example of epipolar lines



This slide shows examples of epipolar lines and corresponding points associated to the image pair.



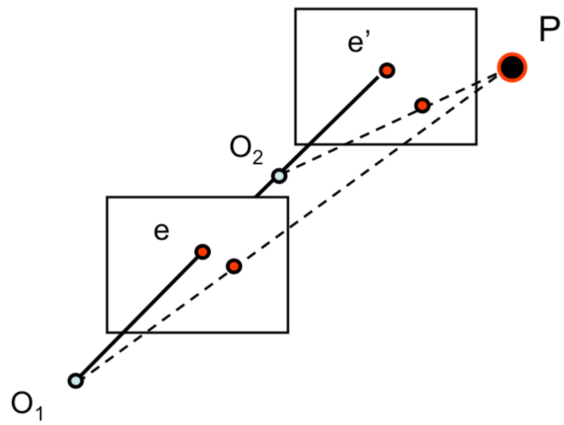
Next we will consider some special cases.

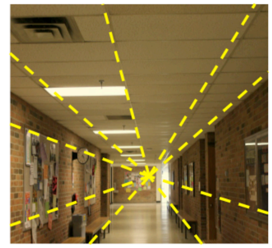
An interesting special case is one in which the image planes are parallel to each other. If the two image planes are parallel to each other, then the epipoles e and e' will be at infinity, since the line joining the centers O_1O_2 (baseline) is parallel to the image planes. An important by product of this is that the epipolar lines are parallel to u axis in each image plane

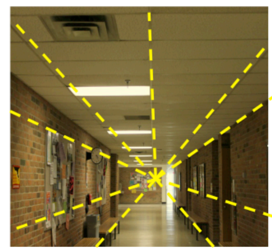
Example: Parallel Image Planes

Here is a practical example of two images whose planes are parallel to each other. The epipoles e and e’ occur at infinity. Notice that the epipolar lines in a given image are (almost) parallel to each other and meet at infinity, where the epipoles occur.

Example: Forward translation

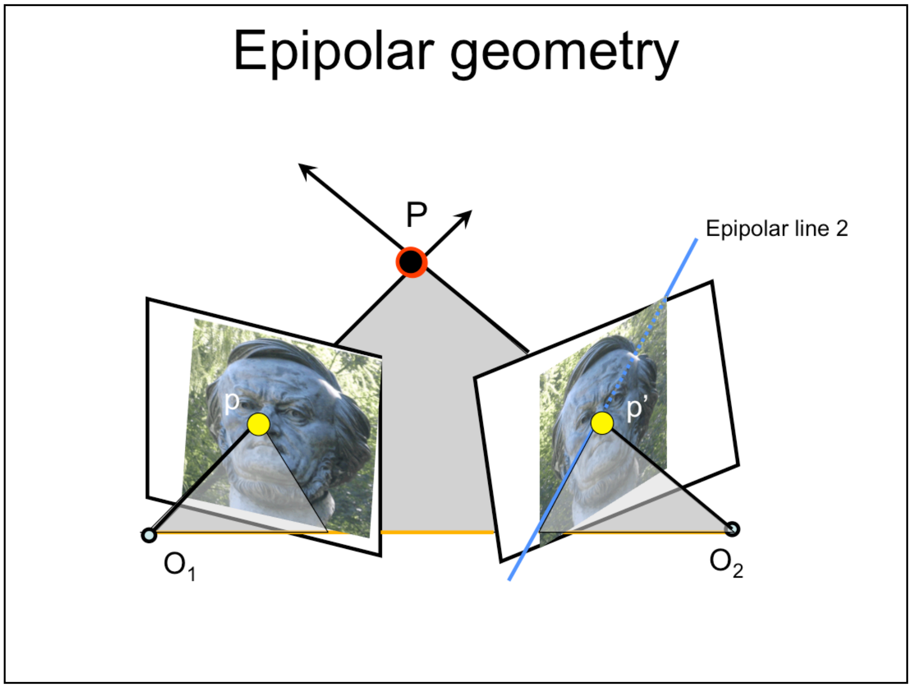






• The epipoles have same position in both images
• Epipole called FOE (focus of expansion)

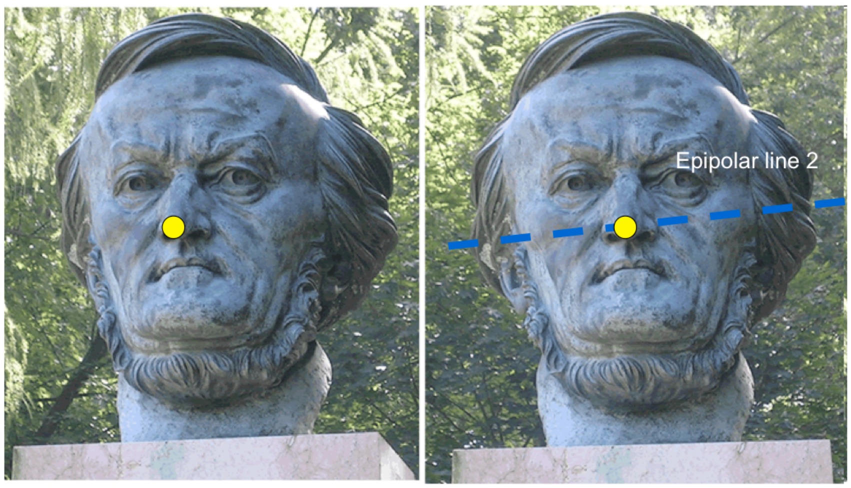
Here is another special case in which there is only a forward translation motion between the two images. In this case, the epipoles will occur at the same location in the two images and are called “focus of expansion”. An example of this configuration is shown to the right.



Assume that the camera positions, orientations and camera matrices are known and suppose that p is the observation of point P in the 3D world coordinate. The epipolar plane is defined by O_1, p and O_2.

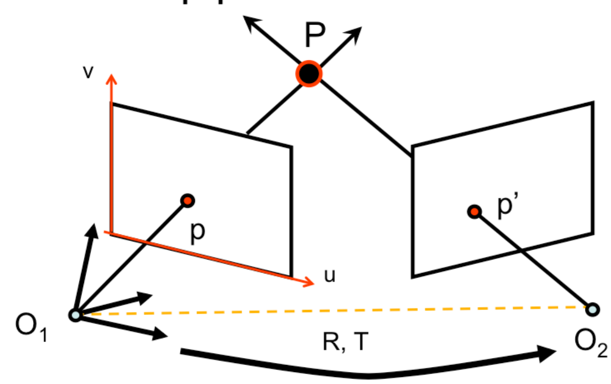
The intersection of the epipolar plane with the camera image planes defines the epipolar lines. Thus, by definition, the corresponding observation p' in the camera should belong to the epipolar line 2. Thus we have established a mapping between p and the second image. Is this mapping one-to-one? No, because we only know that p' must belong this epipolar line and we don't exactly where. But the interesting thing about this example is that epipolar geometry allows us to establish this constraint WITHOUT knowing where P is in 3D! In fact, epipolar lines can found by just using O1, O2 and p!

Epipolar Constraint



Here is the example

Epipolar Constraint



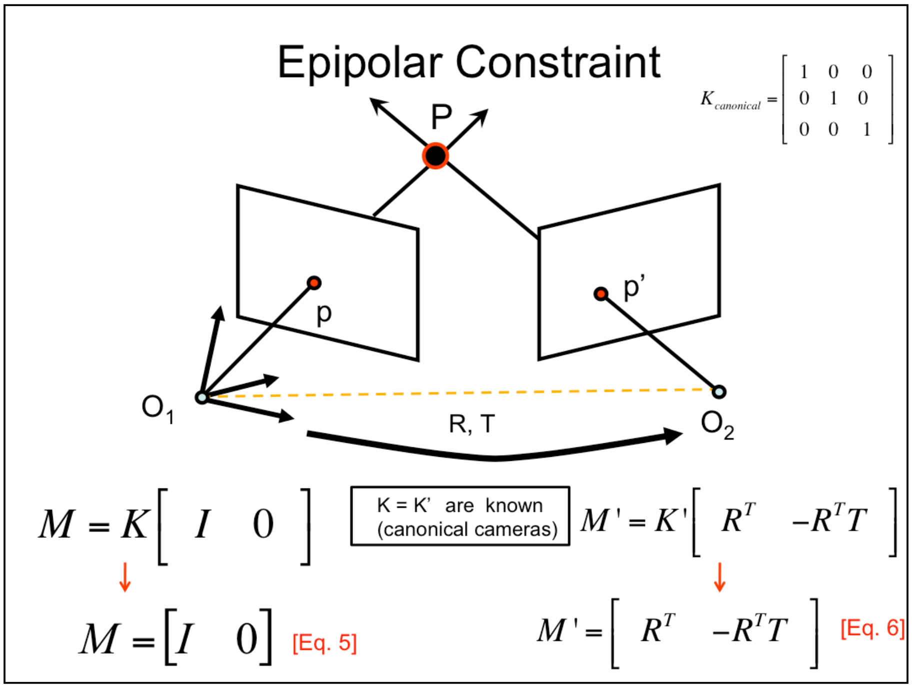
$$M = K\begin{bmatrix} I & 0 \end{bmatrix} \qquad M' = K'\begin{bmatrix} R^T & -R^T T \end{bmatrix}$$

$$M\,P = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = p \quad \text{[Eq. 3]} \qquad M'\,P = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = p' \quad \text{[Eq. 4]}$$

Let us now push this idea further and introduce two important concepts in epipolar geometry: the *essential matrix and fundamental matrix*— these matrices allow to map points and epipolar lines across views. Let M and M' be the projection matrices for camera 1 and 2, respectively. M and M' govern the transformation between 3D world points to 2D image points.

Let us assume that the world reference system is associated to camera 1. Let R and T be the rotation matrix and translation vector that transform the "world" coordinates to the camera 2 coordinates. Under this assumption, M' = K' [R^T – R^T T]; we don't derive this expression here and leave it as an exercise.

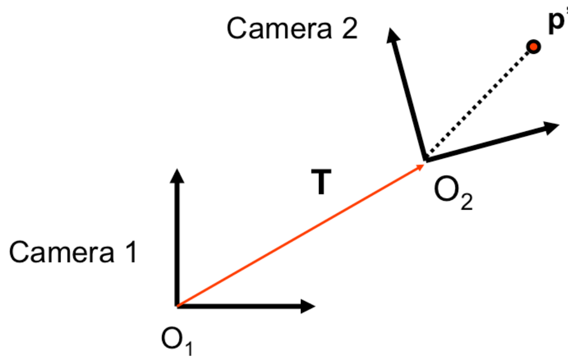
Equations 3 and 4 express this transformation for p and p' (in homogeneous coordinates).



For simplicity, assume now that the cameras are canonical which means that K = K'= I. We'll remove this assumption later on in the lecture. So, the projection matrices can be reduced to the above forms, as given in equations 5 and 6.

Given this simplified expression for M and M', let's try to find a relationship between p and p'.

The cameras are related by R, T



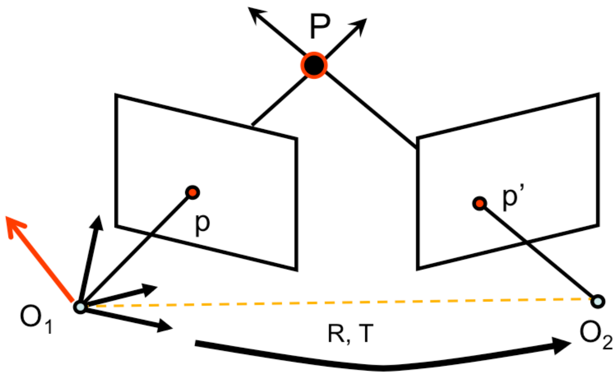
T = $\mathbf{O_2}$ in the camera 1 reference system

R is the rotation matrix such that a vector **p'** in the camera 2 is equal to **R p'** in camera 1.

Assume T is the coordinate vector of the translation O1 O2 separating the two coordinate systems. T gives the translation between the two cameras and R gives the relative rotation. Using T, we can determine the coordinates of the 2nd camera center O2. Since the 1st camera center is assumed to be at the origin (of the world reference system), then the center (O2) of the 2nd camera will be at T.

R is the rotation matrix such that a free vector p' in the camera 2 is equal to R p' in camera 1.

Epipolar Constraint



p’ in first camera reference system is $= R\ p'$

$T \times (R\ p')$ is perpendicular to epipolar plane

$$\rightarrow p^T \cdot \left[T \times (R\ p')\right] = 0 \quad \text{[Eq. 7]}$$

Thus we can express p’ in the first camera references system as R p’.

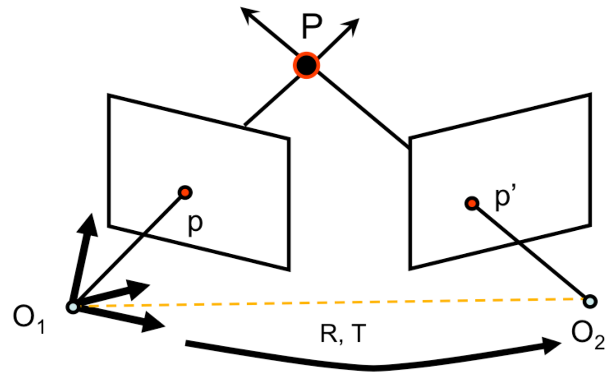
Now notice that T x (Rp’) is perpendicular to the epipolar plane. Since p belongs to the epipolar plane, the dot product of pT and T x (Rp’) is zero which leads us to equation 7.

Cross product as matrix multiplication

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

Let's now introduce a different (more compact) expression for the cross product. This allows us to express the cross-product between two any given vectors **a** and **b** as a matrix-vector multiplication. An example for the case of 3D vectors is illustrated here. In this example, a_x, a_y and a_z (or b_x, b_y and b_z) are the 3 coordinates of **a** (or **b**).

Epipolar Constraint



$$p^T \cdot [T \times (R\, p')] = 0 \rightarrow p^T \cdot \boxed{[T_\times]} \cdot R\, p' = 0$$

[Eq. 8] [Eq. 9]

E = Essential matrix

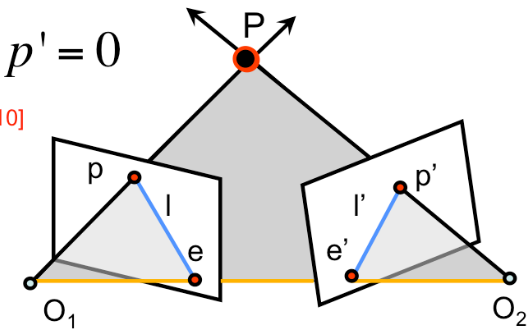
(Longuet-Higgins, 1981)

We can now convert the cross product term into a matrix multiplication and rewrite equation 8 as equation 9. Equation 9 is known as the Epipolar constraint for canonical camera matrices (K=1), and the 3x3 matrix E= $[T_x]$ R is known as the **Essential Matrix.** The essential matrix E was first introduced by Longuet-Higgins in 1981.

Epipolar Constraint

$$p^T \cdot E\, p' = 0$$

[Eq. 10]



- l = E p' is the epipolar line associated with p'
- l' = E^T p is the epipolar line associated with p
- E e' = 0 and E^T e = 0
- E is 3x3 matrix; 5 DOF
- E is singular (rank two)

The Essential matrix and two corresponding points p and p' in the two images satisfy the above Epipolar constraint (given in equation 10). Notice that Eq. 10 is a scalar equation. The Essential matrix is also useful to compute the epipolar lines associated with p and p'. For instance, E p' is the epipolar line (in the image of camera 1) associated with p' in the image of camera 2; that is, l = E p'. Similarly, E^T p is the epipolar line associated with p (l' = E^T p).

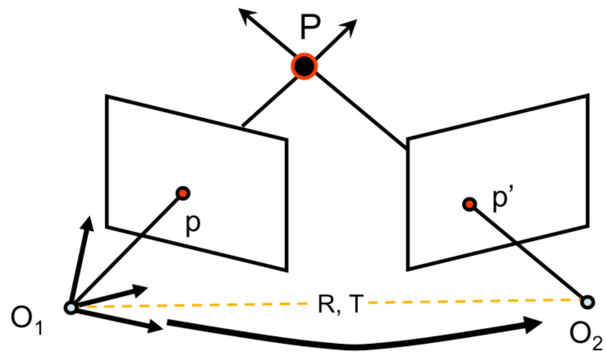
E also satisfies other interesting properties:

The epipoles are solutions of the homogenous equations E e' = 0 and E^T e = 0

E is a 3x3 matrix and has 5 degrees of freedom.

Moreover, E is singular and its rank is 2.

Epipolar Constraint



$$M = K\begin{bmatrix} I & 0 \end{bmatrix} \qquad M' = K'\begin{bmatrix} R^T & -R^T T \end{bmatrix}$$

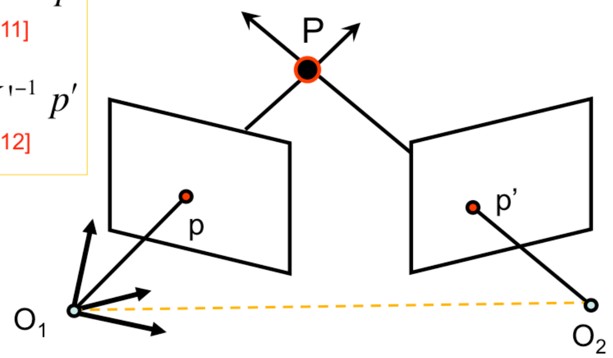
$$p_c = K^{-1} p \quad \text{[Eq. 11]} \qquad p'_c = K'^{-1} p' \quad \text{[Eq. 12]}$$

Now, let's derive a similar relationship between p and p' when the Ks are no longer canonical. So suppose that the cameras are not canonical and let K and K' be the (unknown) camera matrices for camera 1 and 2, respectively. In this case, we can modify the previous derivation to arrive to a slightly different constraint which we define in the next slide. Let p_c and p'_c be the projections of P to the corresponding camera images that we derived by assuming that the camera matrices are canonical. Let p and p' be the projections of P to the corresponding camera images by assuming that the cameras are not canonical with K and K' as defined above. In this case Eq. 11 and 12 are satisfied.

Epipolar Constraint

$$p_c = K^{-1} p$$
[Eq. 11]

$$p'_c = K'^{-1} p'$$
[Eq. 12]



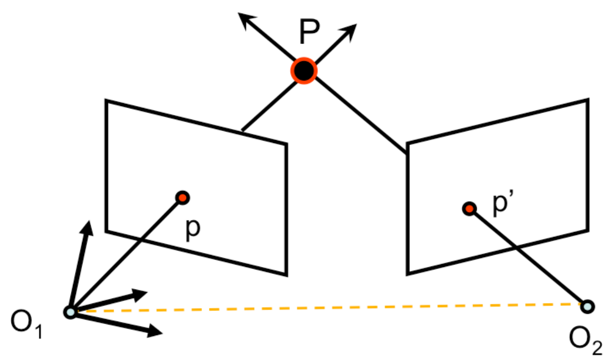
[Eq.9]

$$p_c^T \cdot \left[T_\times\right] \cdot R\, p'_c = 0 \rightarrow (\mathrm{K}^{-1}\, \mathrm{p})^{\mathrm{T}} \cdot \left[\mathrm{T}_\times\right] \cdot \mathrm{R}\, \mathrm{K}'^{-1}\, \mathrm{p}' = 0$$

$$\mathrm{p}^{\mathrm{T}} \boxed{\mathrm{K}^{-\mathrm{T}} \cdot \left[\mathrm{T}_\times\right] \cdot \mathrm{R}\, \mathrm{K}'^{-1}}\, \mathrm{p}' = 0 \rightarrow \mathrm{p}^{\mathrm{T}} \boxed{\mathrm{F}}\, \mathrm{p}' = 0$$
[Eq. 13]

Using Eq. 11 and 12, we replace the expressions of p_c and p'_c in Eq.9 and, by follow the above derivation, we obtain a new epipolar constraint, as given in equation 13.

Epipolar Constraint



[Eq. 13] $\boxed{p^T F p' = 0}$

$$F = K^{-T} \cdot [T_\times] \cdot R\, K'^{-1}$$

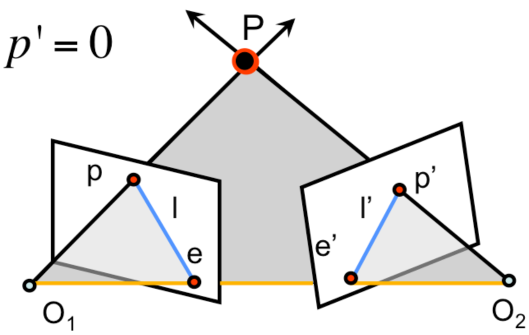
[Eq. 14]

F = Fundamental Matrix
(Faugeras and Luong, 1992)

The epipolar constraint in Eq 13 relates two corresponding points p and p' in the two images by means of a 3x3 matrix F which we define as the **Fundamental Matrix.** The fundamental matrix F (characterized by Eq. 14) encapsulates the parameters from both camera matrices (K and K') as well as the relative translation T and rotation R between them.

Epipolar Constraint

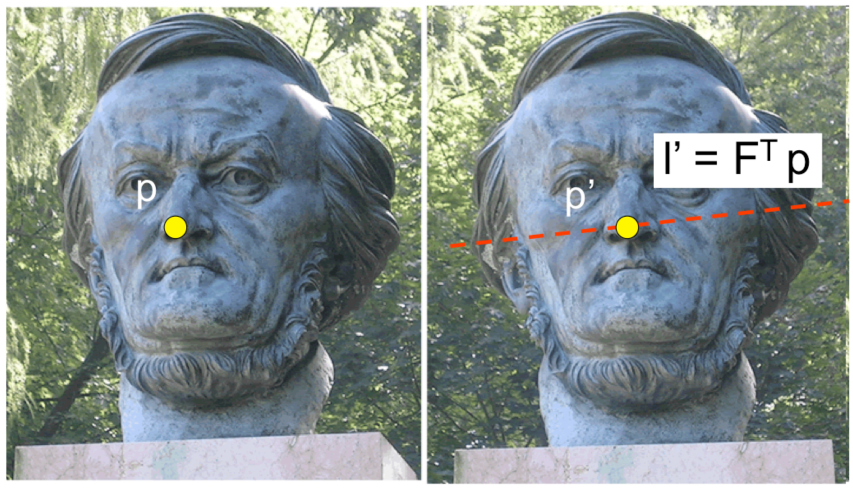
$$p^T \cdot F\ p' = 0$$



- $l = F\ p'$ is the epipolar line associated with p'
- $l' = F^T p$ is the epipolar line associated with p
- $F\ e' = 0$ and $F^T e = 0$
- F is 3x3 matrix; 7 DOF
- F is singular (rank two)

The fundamental matrix is also useful to compute the epipolar lines associated with p and p', when the camera matrices K, K', and the transformation R and T are unknown. These relationships are equivalent to the ones we introduced for the essential matrix. The key difference is that F has 7 degrees of freedom (instead of 5).

Why F is useful?



- Suppose F is known
- No additional information about the scene and camera is given
- Given a point on left image, we can compute the corresponding epipolar line in the second image

Let us revisit the example we discussed earlier:

Given a point p on left image (say the tip of the nose), we know that this will correspond to a epipolar line in the second image which will contain p'. Last time we computed this epipolar line by intersecting the epipolar plane with the second image. Which assumption we made? We assumed that the cameras are calibrated and, thus, that K, R, T are supposed be known. Can we do the same if we don't assume that the cameras are calibrated? Yes! – as long as F is given (known). In that case, for a point p in image 1, we can find the epipolar line ($F^T p$) in image 2 associated with p. The corresponding point in image 2 will lie on this epipolar line, as illustrated on the right hand side.

This procedure shows that even without knowing the actual position of P in 3D and without knowing intrinsic and extrinsic parameters of the cameras, by using the concept of epipolar lines, we can establish a relationship between p and p' (that is, p' the should belong to the epipolar line $F^T p$).

Why F is useful?

- F captures information about the epipolar geometry of 2 views + camera parameters
- MORE IMPORTANTLY: F gives constraints on how the scene changes under view point transformation (without reconstructing the scene!)
- Powerful tool in:
  - 3D reconstruction
  - Multi-view object/scene matching

As noted earlier, F encapsulates information about the cameras as well as the geometry of the 2 views. Without the need for explicit reconstruction of the scene, F gives us a way to capture the viewpoint transformation.

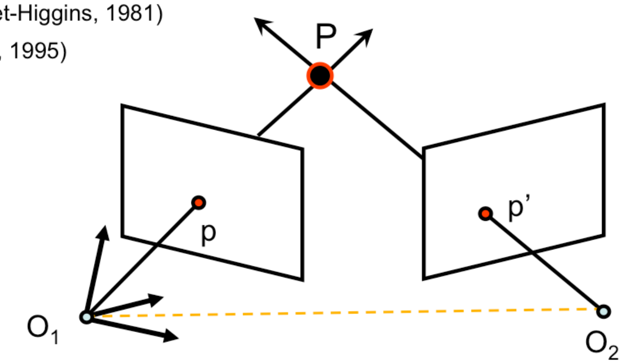
Thus, it is also useful in 3D reconstruction and for establishing correspondences between objects from multiple views.

Estimating F

The Eight-Point Algorithm

(Longuet-Higgins, 1981)

(Hartley, 1995)



$$p^T \, F \, p' = 0$$

Is it possible to estimate F given two images of the same scene and without knowing intrinsic and extrinsic parameters of the cameras? Yes, as long as we have a sufficient number of point correspondences between the two images. There are several algorithms for estimating F.

Next, we will discuss the Eight-Point Algorithm which was initially proposed by Longuet-Higgins in 1981 and further extended by R. Hartley in 1995. This algorithms assume that a set of (at least) 8 pairs of corresponding points between two images is available.

Estimating F

[Eq. 13] $$p^{T}\,F\,p' = 0 \quad \Rightarrow \qquad p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad p' = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

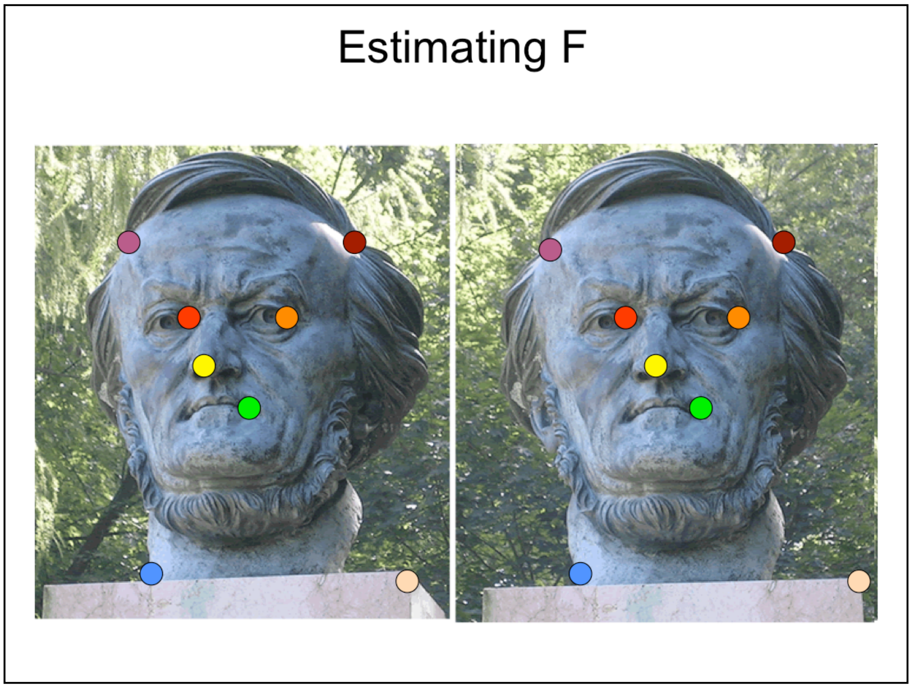
$$(u, v, 1)\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix}\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0$$

$$\Rightarrow \quad (uu', uv', u, vu', vv', v, u', v', 1)\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

[Eq. 14]

Let's take 8 corresponding points

We use the fact that the fundamental matrix satisfies the epipolar constraint (Eq. 13) for all the pairs of corresponding points. Let p and p' be defined as $[u\ v\ 1]^T$ and $[u'\ v'\ 1]^T$ respectively. The constraint in Eq 13 can be rewritten as in Eq. 14. Let's now consider 8 pairs of corresponding points.



This illustration depicts an example of 8 pairs of corresponding points across these two images. Corresponding points are shown with the same color.

Estimating F

$$\left(u_i u'_i, u_i v'_i, u_i, v_i u'_i, v_i v'_i, v_i, u'_i, v_i', 1\right) \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0 \qquad \text{[Eq. 14]}$$

Each correspondence gives us a constraint (Eq. 14).

Estimating F

$$\mathbf{W} \quad \begin{pmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 & 1 \\ u_2u'_2 & u_2v'_2 & u_2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 & 1 \\ u_3u'_3 & u_3v'_3 & u_3 & v_3u'_3 & v_3v'_3 & v_3 & u'_3 & v'_3 & 1 \\ u_4u'_4 & u_4v'_4 & u_4 & v_4u'_4 & v_4v'_4 & v_4 & u'_4 & v'_4 & 1 \\ u_5u'_5 & u_5v'_5 & u_5 & v_5u'_5 & v_5v'_5 & v_5 & u'_5 & v'_5 & 1 \\ u_6u'_6 & u_6v'_6 & u_6 & v_6u'_6 & v_6v'_6 & v_6 & u'_6 & v'_6 & 1 \\ u_7u'_7 & u_7v'_7 & u_7 & v_7u'_7 & v_7v'_7 & v_7 & u'_7 & v'_7 & 1 \\ u_8u'_8 & u_8v'_8 & u_8 & v_8u'_8 & v_8v'_8 & v_8 & u'_8 & v'_8 & 1 \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0 \quad \mathbf{f}$$

[Eqs. 15]

- Homogeneous system $\mathbf{W}\,\mathbf{f} = 0$
- Rank 8 $\longrightarrow$ A non-zero solution exists (unique)
- If N>8 $\longrightarrow$ Lsq. solution by SVD! $\longrightarrow$ $\hat{\mathrm{F}}$

  $\|\mathbf{f}\| = 1$

These constraints can be rearranged to form the homogeneous system in Eqs 15 which can be compactly described as **W f** = 0. **W** is a matrix that collects the coordinates u_i, v_i of p_i in image 1 and the coordinates u_i', v_i' of p_i' in image 2, for i=1 to 8. So **W** is a matrix of measurements. **f** is a vector that collects all the (unknown) coefficients of the matrix F.

The solution to this system of homogeneous equations can be found in the Least Squares sense by Singular Value Decomposition (since Rank(**W**)=8, which means **W** is rank deficient). We enforce the constraint that the norm of the solution is unitary (to avoid finding the trivial zero solution). Let F^{hat} to be the solution of such system.

$\hat{F}$ satisfies: $p^T \hat{F} p' = 0$

and estimated $\hat{F}$ may have full rank ($\det(\hat{F}) \neq 0$)

But remember: fundamental matrix is Rank2

Find F that minimizes $\left\| F - \hat{F} \right\| = 0$

Frobenius norm (*)

Subject to det(F)=0

SVD (again!) can be used to solve this problem

(*) Sq. root of the sum of squares of all entries

Obviously, F^{hat} satisfies the epipolar constraint in Eq. 13 for every pair of corresponding points p and p'. However, F^{hat} is not necessarily a *proper* fundamental matrix. In fact, the fundamental matrix is a rank 2 matrix, but our solution (using SVD) does not guarantee that it is a rank 2 matrix. So, we look for a solution that is the best rank-2 approximation of F^{hat}. In details, we seek an F that minimizes the Frobenius norm of $F-F^{hat}$ subject to the constraint that det(F)=0.

The Frobenius norm of a matrix is the square root of the sum of squares of all entries of the matrix. The solution of the above problem can obtained by SVD again.

Find F that minimizes $\left\| \mathrm{F} - \hat{\mathrm{F}} \right\| = 0$

Frobenius norm (*)

Subject to det(F)=0

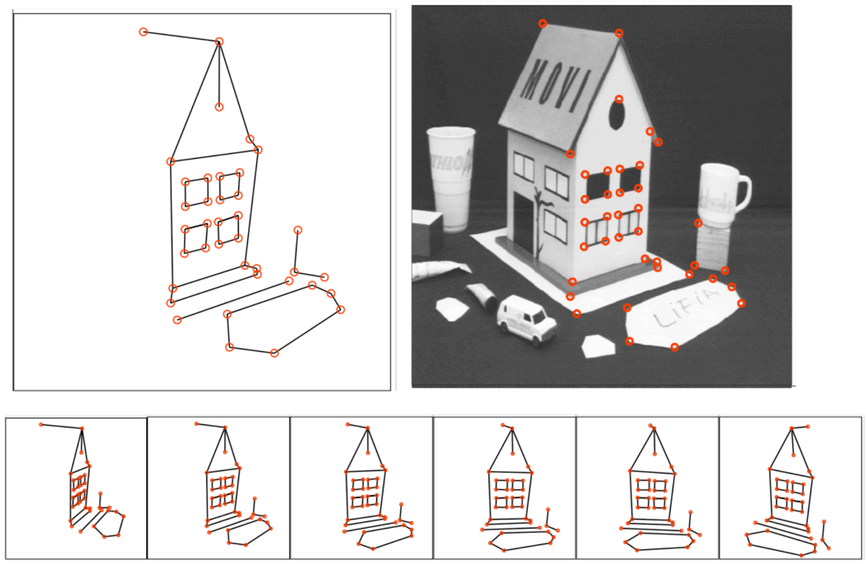
$$F = U \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

Where:

$$U \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & s_3 \end{bmatrix} V^T = SVD(\hat{F})$$

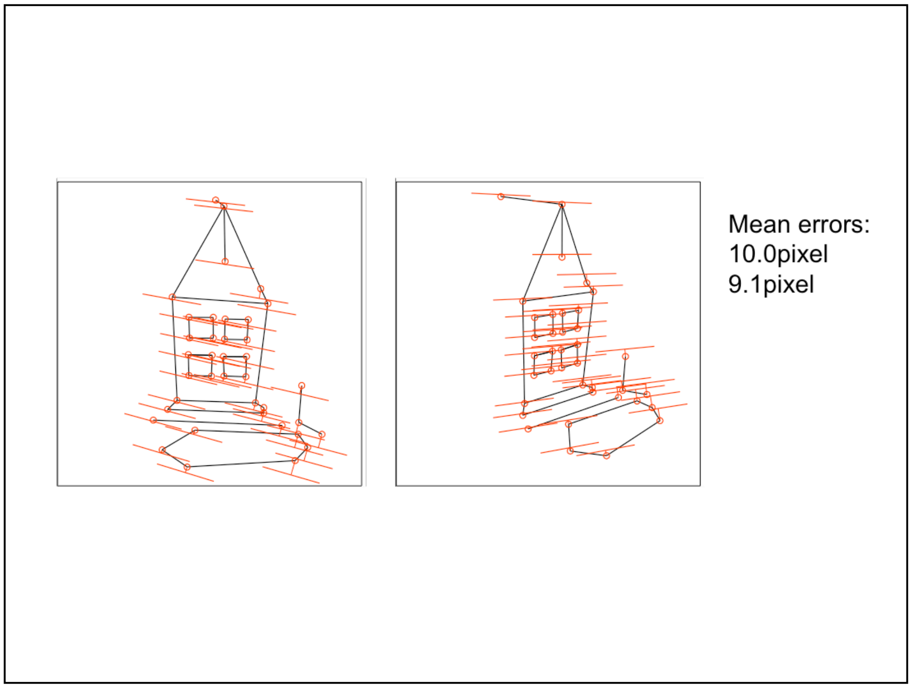
[HZ] pag 281, chapter 11, "Computation of F"

In a few more details, the solution of the above problem can be obtained as U Diag (s1, s2, 0) V^T where U diag (s1, s2, s3) V^T = SVD (F^{hat}), with s1 >= s2 >= s3; See [HZ] pag 281, chapter 11 for details.



Data courtesy of R. Mohr and B. Boufama.

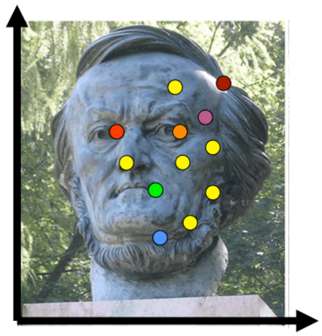
Here we see an example. Given a set of corresponding pairs of points, we can use the 8-point algorithm to determine F.



Once we know F, we can also plot the corresponding epipolar lines of each of the points in the two images. We can compute the mean error (distance between epipolar line and the corresponding point) in terms of pixels.

Notice a rather significantly large mean error (~10 pixels). To reduce this error, we can consider a modified version of the 8-point algorithm which is called the "Normalized Eight-Point Algorithm" as we will explain next.

Problems with the 8-Point Algorithm



$$\mathbf{W}\,\mathbf{f} = 0,\quad \|\mathbf{f}\| = 1 \xrightarrow{\text{Lsq solution by SVD}} \mathrm{F}$$

- Recall the structure of W:
  - do we see any potential (numerical) issue?

What’s the problem with the current version of the Normalized Eight-Point Algorithm? Notice that the constraint under which W f is minimized is not invariant under similarity transformation. Recall the structure of W: do we see any potential (numerical) issue?

Problems with the 8-Point Algorithm

$$\mathbf{Wf} = 0$$

$$\begin{pmatrix} u_1u'_1 & u_1v'_1 & u_1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 & 1 \\ u_2u'_2 & u_2v'_2 & u_2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 & 1 \\ u_3u'_3 & u_3v'_3 & u_3 & v_3u'_3 & v_3v'_3 & v_3 & u'_3 & v'_3 & 1 \\ u_4u'_4 & u_4v'_4 & u_4 & v_4u'_4 & v_4v'_4 & v_4 & u'_4 & v'_4 & 1 \\ u_5u'_5 & u_5v'_5 & u_5 & v_5u'_5 & v_5v'_5 & v_5 & u'_5 & v'_5 & 1 \\ u_6u'_6 & u_6v'_6 & u_6 & v_6u'_6 & v_6v'_6 & v_6 & u'_6 & v'_6 & 1 \\ u_7u'_7 & u_7v'_7 & u_7 & v_7u'_7 & v_7v'_7 & v_7 & u'_7 & v'_7 & 1 \\ u_8u'_8 & u_8v'_8 & u_8 & v_8u'_8 & v_8v'_8 & v_8 & u'_8 & v'_8 & 1 \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

- Highly un-balanced (not well conditioned)
- Values of W must have similar magnitude
- This creates problems during the SVD decomposition

HZ pag 108

Yes! **W** is not a well conditioned matrix (un-balanced) due to the fact that the values of **W** are not necessarily in the same order of magnitude. This creates problems during the SVD decomposition. See HZ pag 108 for details.

Normalization

IDEA: Transform image coordinates such that the matrix **W** becomes better conditioned (**pre-conditioning**)

For each image, apply a following transformation T (translation and scaling) acting on image coordinates such that:

- Origin = centroid of image points
- Mean square distance of the image points from origin is ~2 pixels

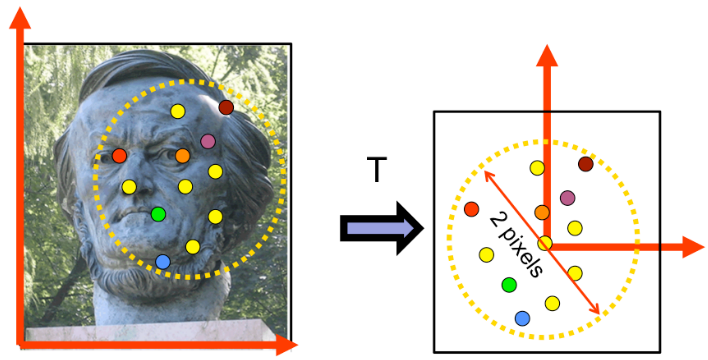
A way to fix this is to normalize the points in the image before solving for **W f** = 0. That is, transform the image coordinates such that the matrix **W** becomes better conditioned (**pre-conditioning**).

Image points are normalized by applying a transformation T (translation and scaling) acting on image coordinates such that:

- The origin of the new coordinate system is located at the centroid of the image points (in the old coordinate system)

- The mean square distance of the transformed image points from origin is ~2 pixels

Example of normalization



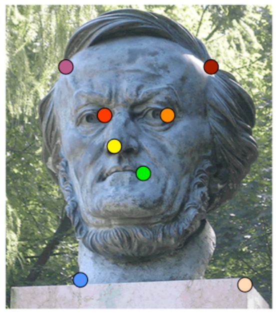
Coordinate system of the image before applying T

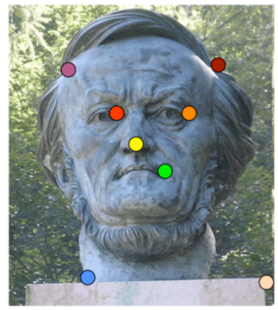
Coordinate system of the image after applying T

- Origin = centroid of image points
- Mean square distance of the image points from origin is ~2 pixels

Here is an example of the effect of T on the image points (for one of the images in the pair).

Normalization





$$q_i = T \; p_i \qquad q_i' = T' \; p_i'$$

So we apply a transformation T on image 1 and a (possible) different transformation T' on image 2.

Let us call q_i and q_i' the normalized points p_i and p'_I in image 1 and 2, respectively.

After that, we apply the same procedure as before on q_i and q_i' for i=1 to 8.

The Normalized Eight-Point Algorithm

0. Compute T and T' for image 1 and 2, respectively

1. Normalize coordinates in images 1 and 2:

$$q_i = T\ p_i \qquad q_i' = T'\ p_i'$$

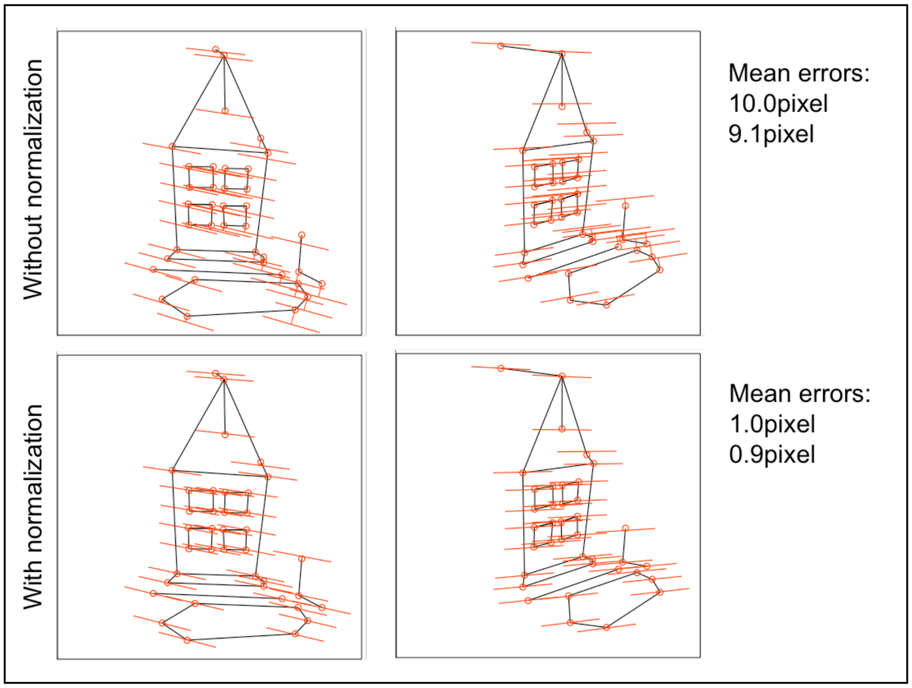
2. Use the eight-point algorithm to compute $\hat{F}_q$ from the corresponding points q_i and q_i'.

1. Enforce the rank-2 constraint. $\rightarrow F_q$ such that:

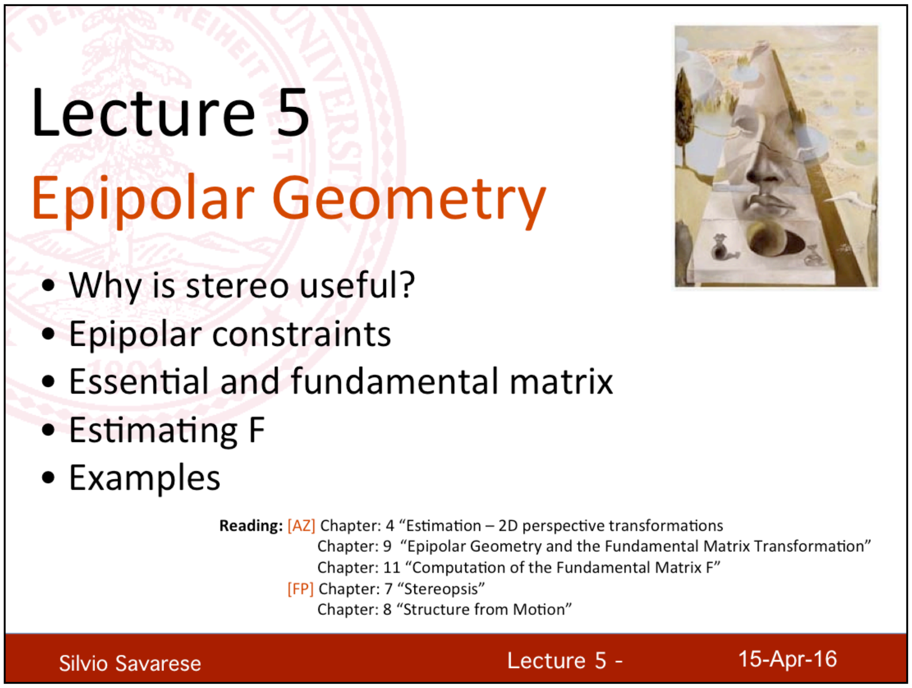
$$\begin{cases} q^T F_q q' = 0 \\ \det(F_q) = 0 \end{cases}$$

2. De-normalize F_q: $F = T^T F_q T'$

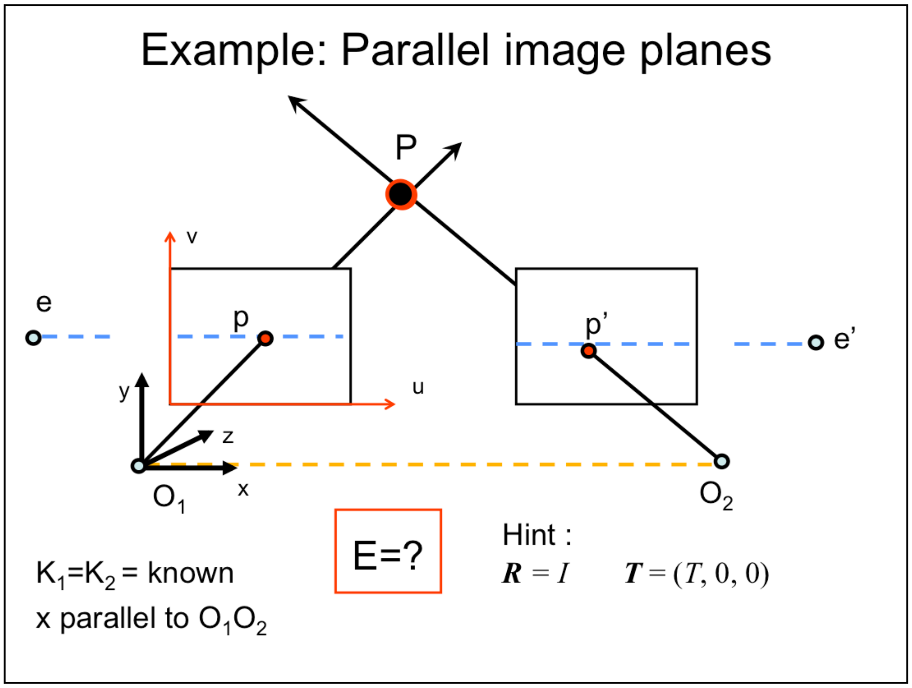
Thus, the modified version of the Eight-Point Algorithm is summarized above. Once F_q^{hat} is computed and the rank-2 constraint is enforced, the matrix F_q is de-normalized by inverting the transformation , and F is obtained as: $F = T^T F_q T'$.



Notice the mean error has been significantly reduced.



In the next slides we will show an example E and F for the special case that images are parallel.



Let us try to compute the essential matrix E in the case of parallel image planes. We assume that the two cameras have the same K, and that K is known. Since the image planes are parallel, there is no relative rotation between the two cameras. So, R = I. There is only a translation T, along the x-axis. So, **T** = (T,0,0). With this information, we set out to find E…

Essential matrix for parallel images

$$\mathbf{E} = [\mathbf{T}_\times] \cdot \mathbf{R}$$

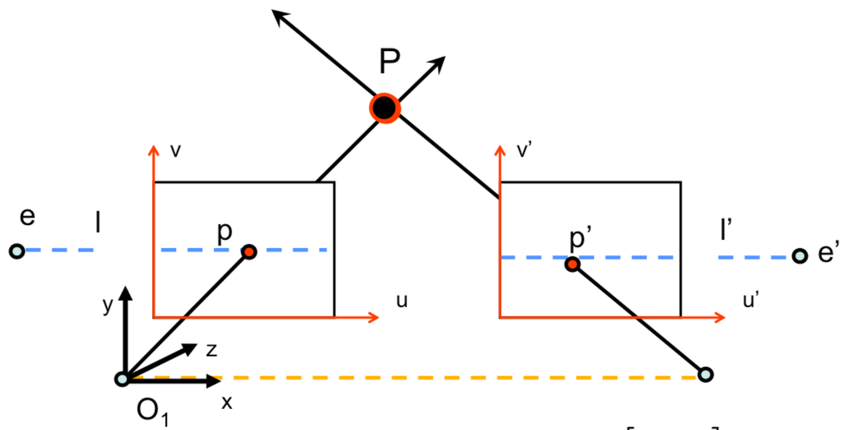
$$\mathbf{E} = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \mathbf{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}$$

[Eq. 20]

$\mathbf{T} = [\ \mathrm{T}\ \ 0\ \ 0\ ]$
$\mathbf{R} = \mathbf{I}$

Recall that E can be computed as the cross product of T and R. Using matrix multiplication to compute the cross product (as described in the previous slides), it is easy to reduce E to the form in equation 20.

Example: Parallel image planes

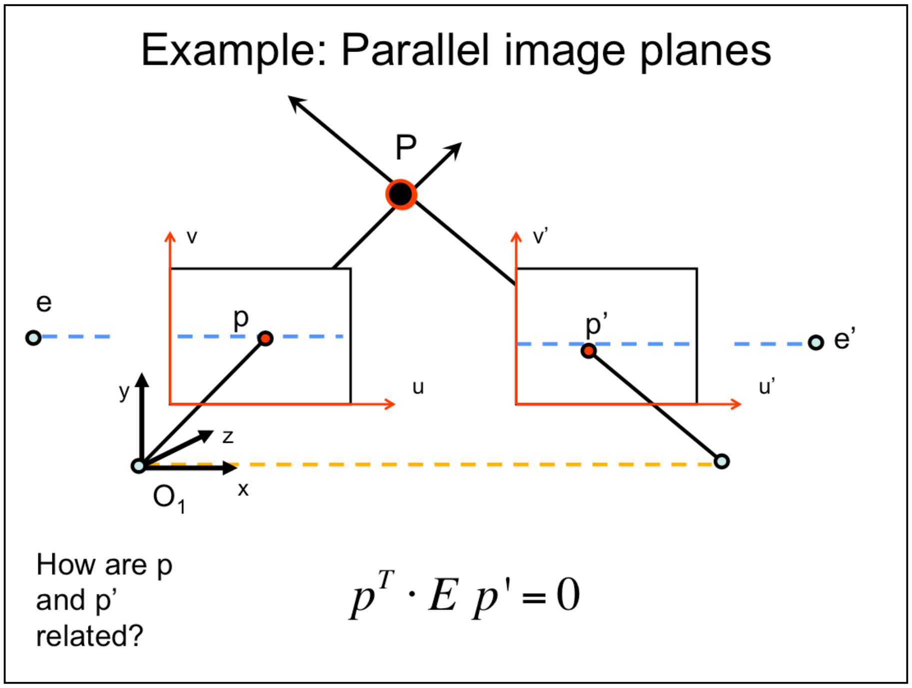


What are the directions of epipolar lines?

$$l = \mathrm{E}\, p' = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -T \\ T v' \end{bmatrix}$$

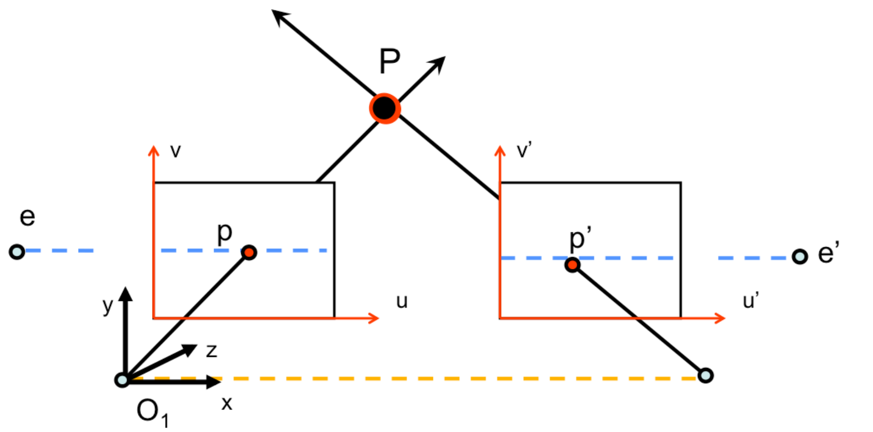
horizontal!

Once E is known, we can find the directions of the epipolar lines associated with points in the image planes. Let us compute the direction of the epipolar line l associated with point p'. The direction of the epipolar line l = E p'. We can see that the direction of l is horizontal, as we would expect to see in this case. We can perform E p to compute l', which can also be verified to be horizontal.



As we have seen before, we can relate p and p' through the Epipolar constraint.

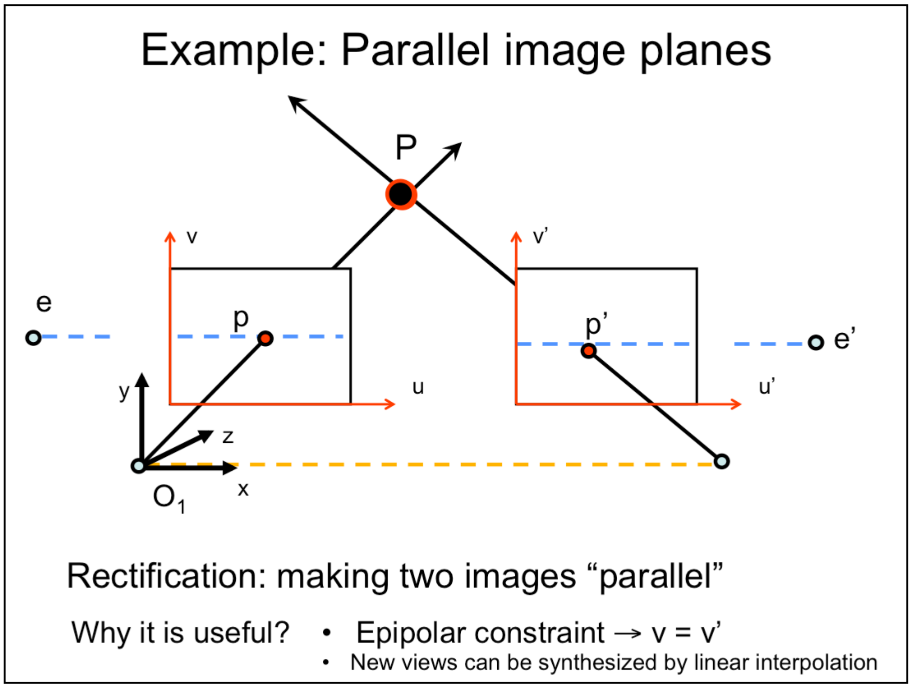
Example: Parallel image planes



How are p and p' related?

$$\Rightarrow \begin{pmatrix} u & v & 1 \end{pmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix} \begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} = 0 \Rightarrow \begin{pmatrix} u & v & 1 \end{pmatrix} \begin{pmatrix} 0 \\ -T \\ Tv' \end{pmatrix} = 0 \Rightarrow Tv = Tv' \Rightarrow v = v'$$

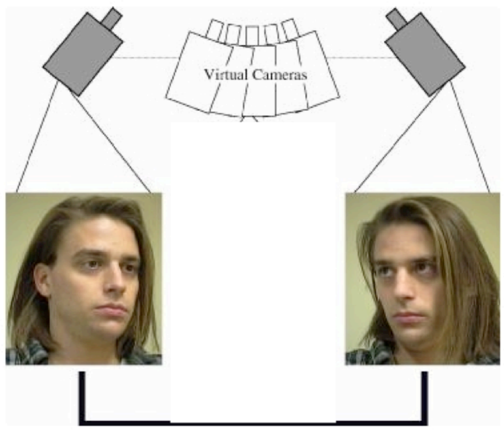
We can use the Epipolar constraint to relate the y coordinates of p and p'. As the slide shows, v = v' which implies that p and p' share the same v-coordinate as expected.



Rectification is the process of making two given images parallel. It is useful because there is a straightforward relationship between corresponding points (they share the same v coordinate) when two images are rectified; also, it possible to synthesize novel intermediate views by simple linear interpolation of the two parallel images. This technique is called **view morphing** as explained next.

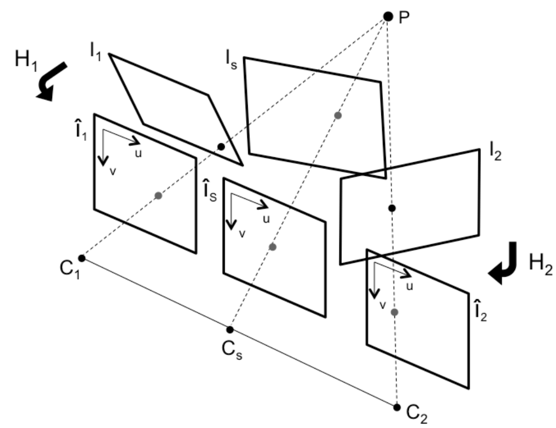
Application: view morphing

S. M. Seitz and C. R. Dyer, *Proc. SIGGRAPH 96*, 1996, 21-30

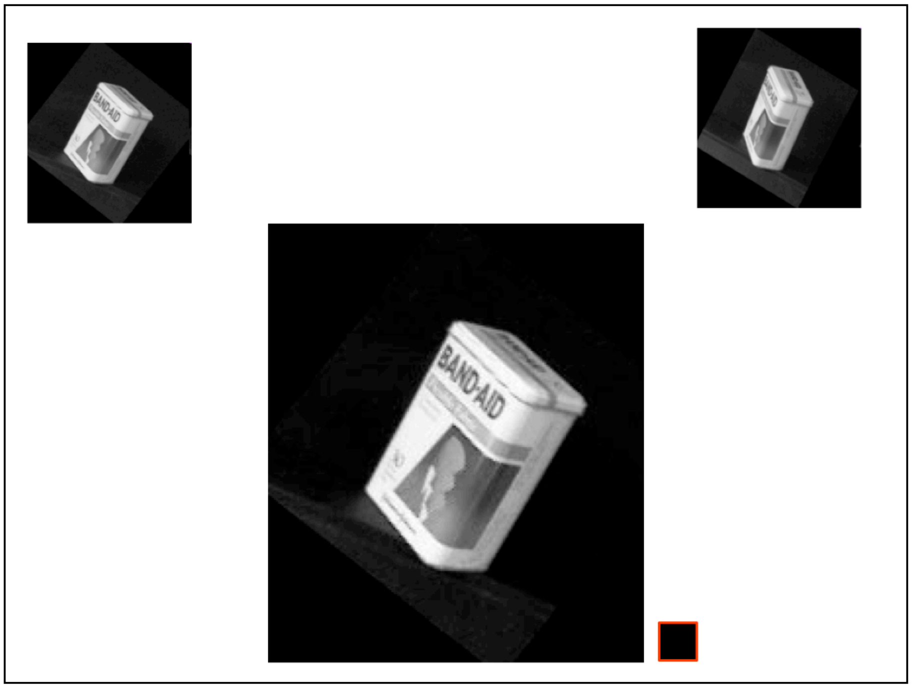


This technique was first introduced by Seitz and Dyer in 1996. We see an example here.

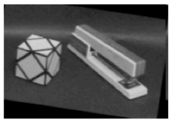
Rectification

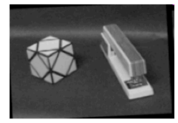


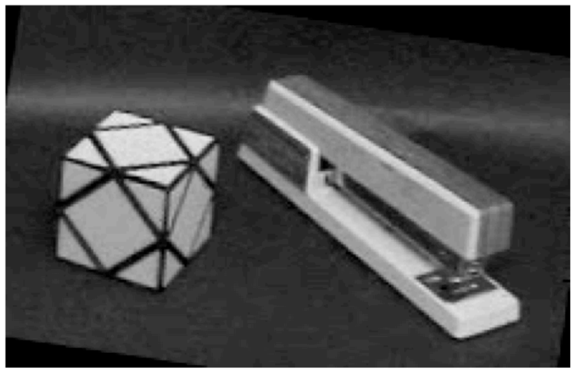
In view morphing, we first convert the two images into the parallel configuration. This step is known as pre-warping, and is done using two homographies H_1 and H_2 for images I_1 and I_2. We then linearly interpolate positions and intensities of corresponding pixels in I_1 and I_2 to form I_s. Finally, we post-warp I_s using a homography H_s to obtain the final desired image.

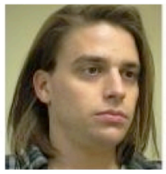


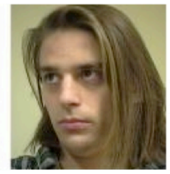
Here is we see some example of view morphing.

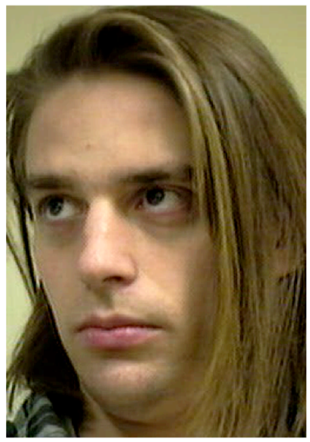


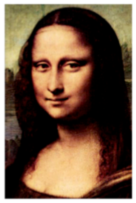


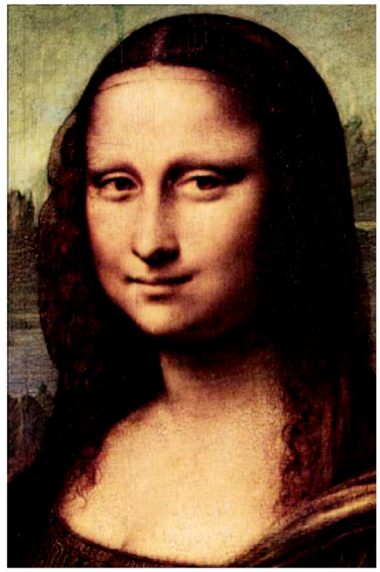


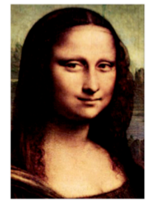




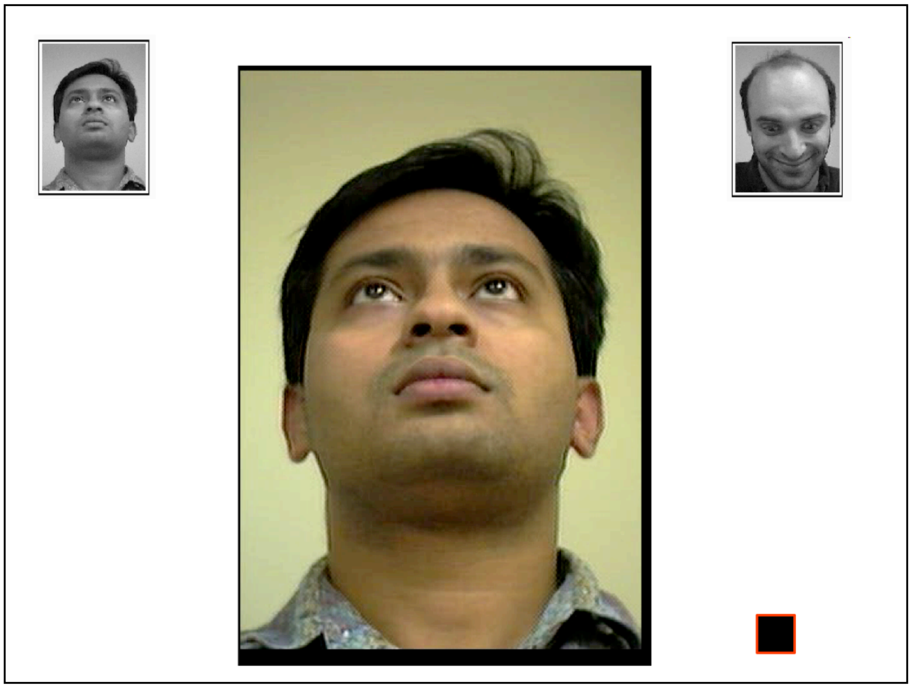








From its reflection!



The Fundamental Matrix Song