

Image-based Modeling and Rendering

2. Image Warping & Morphing (B)

Course no. ILE5025

National Chiao Tung Univ, Taiwan

By: I-Chen Lin, Assistant Professor

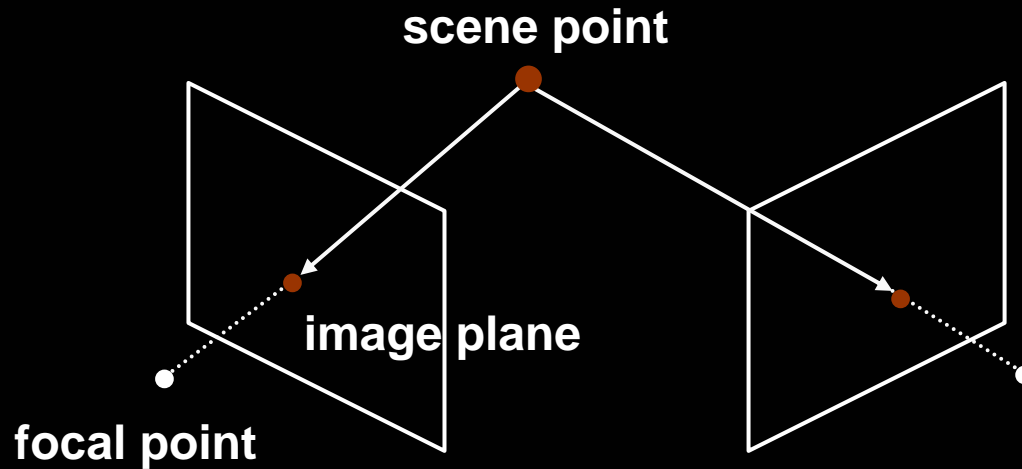
Outline

- Epipolar Geometry
- Estimating prewarping matrices.

Reference:

- Prof. S. Seitz and P. Heckbert (CMU), Image-based modeling and rendering course notes.
- S. M. Seitz, C. R. Dyer, "View Morphing", Proc. SIGGRAPH'96, pp. 21-30.
- R.I. Hartley, "In defence of the 8-point algorithm", Proc. Intl. Conf. Computer Vision (ICCV'95), pp.1064-1070, 1995.
- Dr. Ng Teck Khim, Computer Vision and Graphics for Special Effects lecture notes.

Two-View Projective Geometry

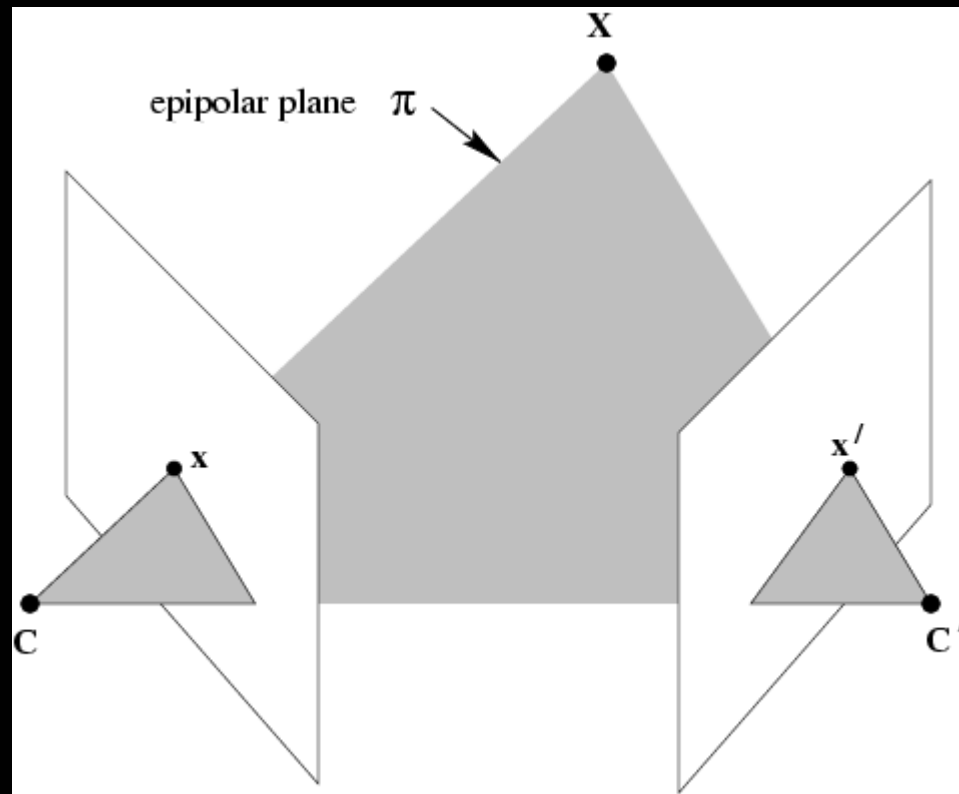


- How to relate point positions in different views?
 - Central question in image-based rendering
 - Projective geometry gives us some powerful tools
 - constraints between two or more images
 - equations to transfer points from one image to another

Two-View Projective Geometry

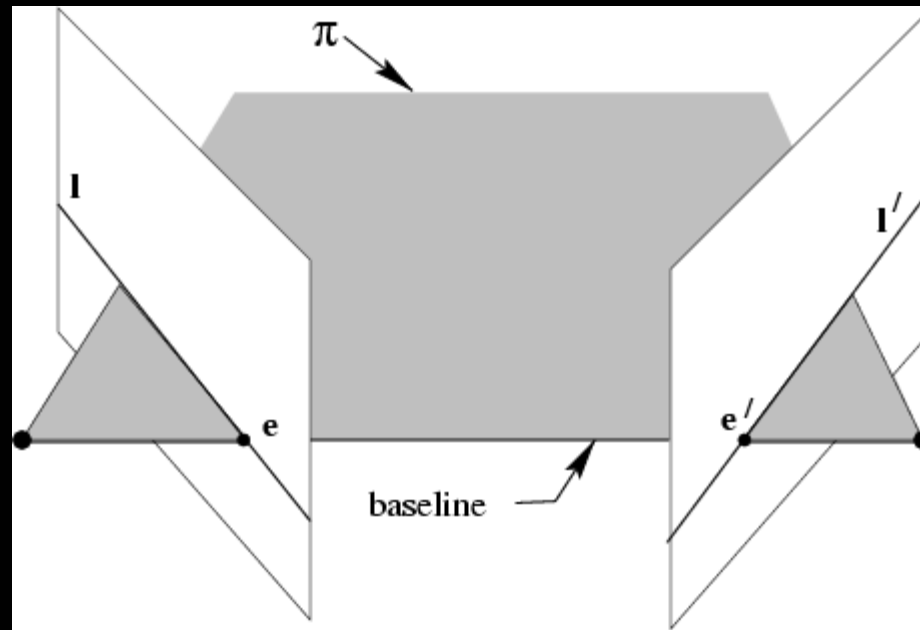
- Correspondence geometry:
 - Given an image point x in the first view, how does this constrain the position of the corresponding point x' in the second image?
- Camera geometry (motion):
 - Given a set of corresponding image points $\{x_i \leftrightarrow x'_i\}$, $i=1, \dots, n$, what are the cameras C and C' for the two views? Or what is the geometric transformation between the views?
- Scene geometry (structure):
 - Given corresponding image points $x_i \leftrightarrow x'_i$ and cameras C, C' , what is the position of the point X in space?

Epipolar Geometry



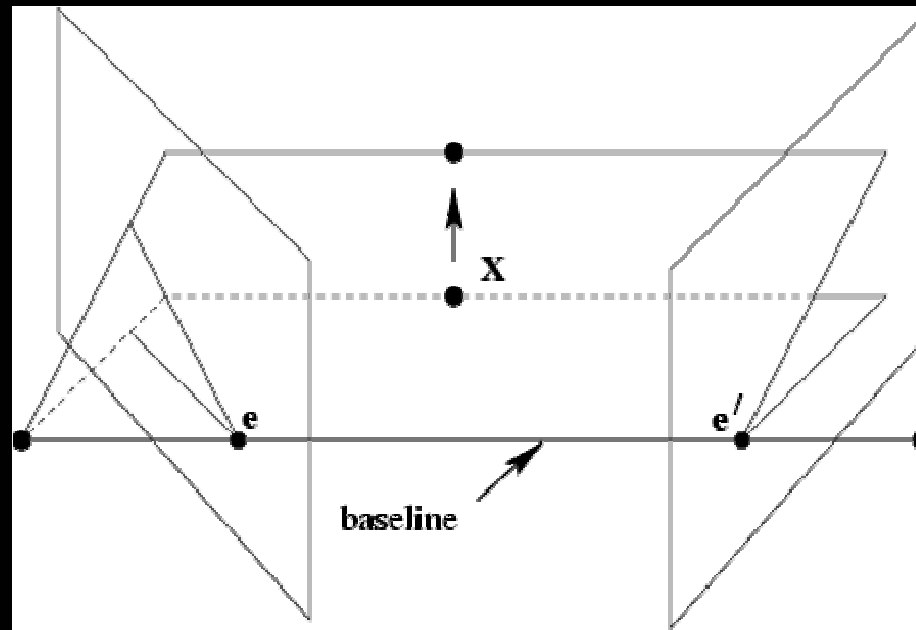
C, C', x, x' and X are coplanar

Epipolar Geometry



All points on π project on l and l'

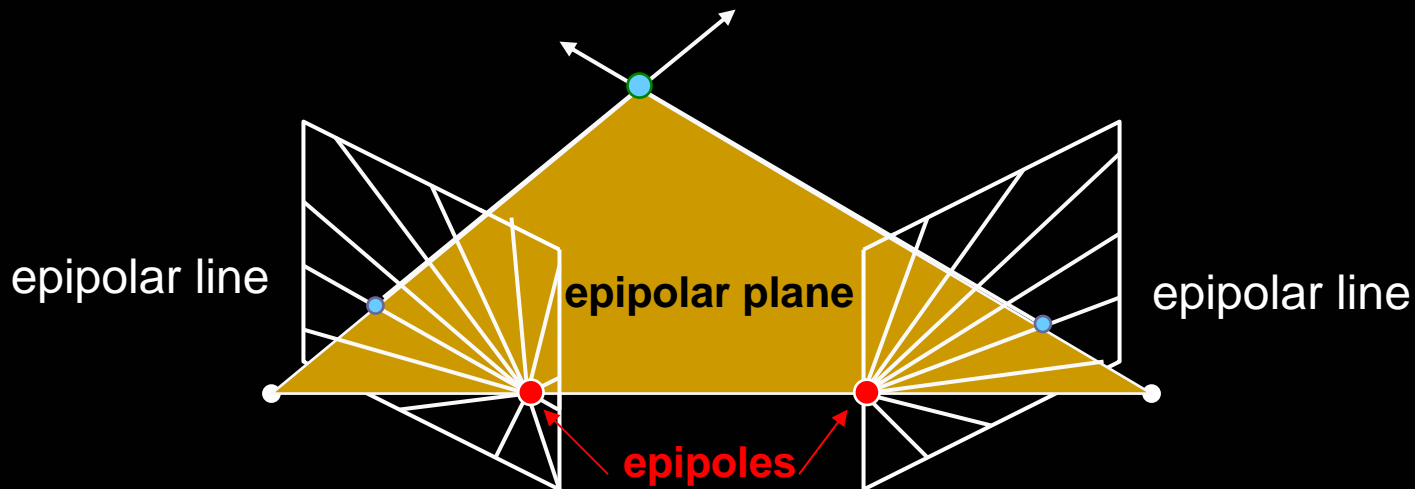
Epipolar Geometry



Family of planes π and lines l and l'
Intersection in e and e'

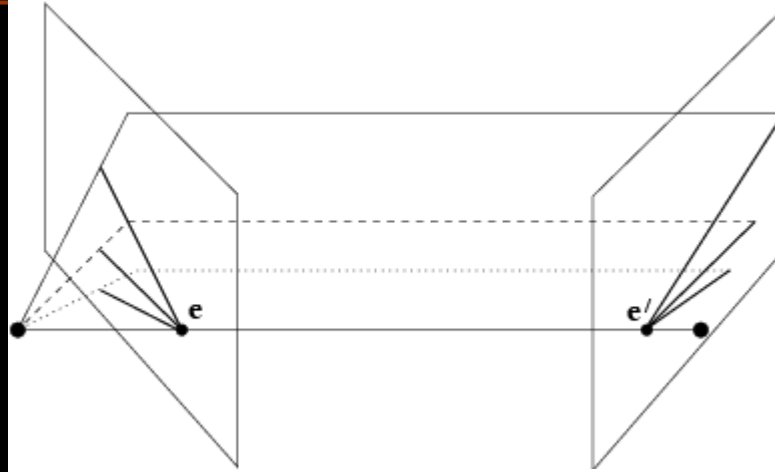
Epipolar Geometry

- What does one view tell us about another?
 - Point positions in 2nd view must lie along a known line

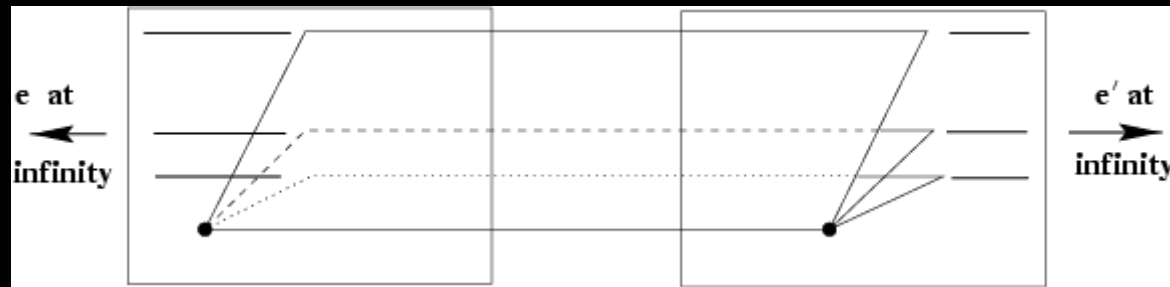


- Epipolar Constraint
 - Extremely useful for stereo matching
 - Reduces problem to 1D search along *conjugate epipolar lines*
 - Also useful for view interpolation...

Example: Converging Cameras



Example: Motion Parallel with Image Plane



3D to 2D: Perspective Projection

Matrix Projection:

$$\mathbf{p} = \begin{bmatrix} sx \\ sy \\ s \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi P}$$

$\mathbf{\Pi}$ can be decomposed into $\mathbf{T} \rightarrow \mathbf{R} \rightarrow \text{project} \rightarrow \mathbf{A}$

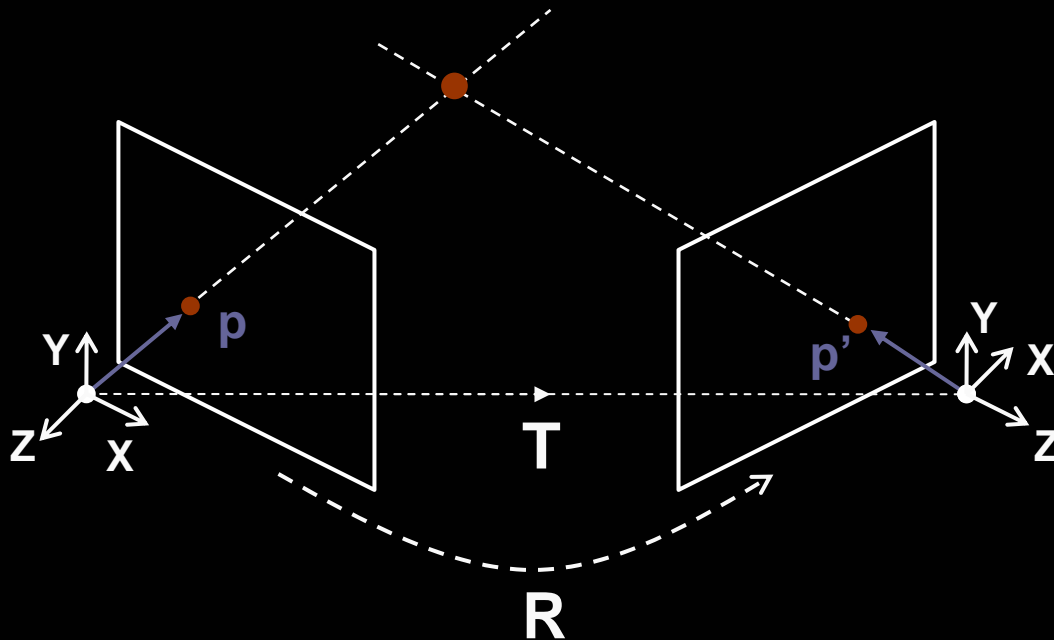
$$\mathbf{\Pi} = \begin{bmatrix} s_x & 0 & -t_x \\ 0 & s_y & -t_y \\ 0 & 0 & 1/f \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{T}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$$

Then we can write the projection as:

$$\mathbf{p} = \mathbf{\Pi P} = \mathbf{AR}(\mathbf{P} + \mathbf{T})$$

Epipolar Algebra

- How do we compute epipolar lines?
 - Can trace out lines, reproject. But that is overkill



$$\mathbf{p}' = \mathbf{R}\mathbf{p} + \mathbf{T}$$

- Note that \mathbf{p}' is \perp to $\mathbf{T} \times \mathbf{p}'$
 - So $0 = \mathbf{p}'^T \mathbf{T} \times \mathbf{p}' = \mathbf{p}'^T \mathbf{T} \times (\mathbf{R}\mathbf{p} + \mathbf{T}) = \mathbf{p}'^T \mathbf{T} \times (\mathbf{R}\mathbf{p})$

Simplifying: $\mathbf{p}'^T \mathbf{T} \times (\mathbf{R}\mathbf{p}) = 0$

- We can write a cross-product $\mathbf{a} \times \mathbf{b}$ as a matrix equation

- $\mathbf{a} \times \mathbf{b} = \mathbf{A}_{\times} \mathbf{b}$ where

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\times} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ y & x & 0 \end{bmatrix}$$

- Therefore: $\mathbf{0} = \mathbf{p}'^T \mathbf{E} \mathbf{p}$

- Where $\mathbf{E} = \mathbf{T}_{\times} \mathbf{R}$ is the 3x3 "essential matrix"
- Holds whenever \mathbf{p} and \mathbf{p}' correspond to the same scene point

Simplifying: $\mathbf{p}'^T \mathbf{T} \times (\mathbf{R} \mathbf{p}) = \mathbf{0}$

■ Properties of E

- $E \mathbf{p}$ is the epipolar line of \mathbf{p} ; $\mathbf{p}'^T E$ is the epipolar line of \mathbf{p}'
 - $\mathbf{p}'^T E \mathbf{p} = 0$ for every pair of corresponding points
 - $\mathbf{0} = E \mathbf{e} = \mathbf{e}'^T E$ where \mathbf{e} and \mathbf{e}' are the epipoles
 - E has rank < 3 , has 5 independent parameters
- E tells us *everything* about the epipolar geometry

Linear Multiview Relations

- The Essential Matrix: $0 = p'^T E p$
 - First derived by Longuet-Higgins, Nature 1981
 - also showed how to compute camera R and T matrices from E
 - E has only 5 free parameters (three rotation angles, two transl. directions)
 - Only applies when cameras have same internal parameters
 - same focal length, aspect ratio, and image center

Linear Multiview Relations (cont.)

- The Fundamental Matrix: $0 = \mathbf{p}'^T \mathbf{F} \mathbf{p}$
 - $\mathbf{F} = (\mathbf{A}'^{-1})^T \mathbf{E} \mathbf{A}^{-1}$, where $\mathbf{A}_{3 \times 3}$ and $\mathbf{A}'_{3 \times 3}$ contain the internal parameters
 - Gives epipoles, epipolar lines
- \mathbf{F} (like \mathbf{E}) is defined only up to a scale factor and has rank 2 (7 free params) [There are 9 elements, but scaling is not significant and $\det \mathbf{F} = 0$]
 - Generalization of the essential matrix
 - Can't uniquely solve for \mathbf{R} and \mathbf{T} (or \mathbf{A} and \mathbf{A}') from \mathbf{F}
 - Can be computed using linear methods
 - R. Hartley, *In Defence of the 8-point Algorithm*, ICCV 95
 - Or nonlinear methods

$$\mathbf{A} = \begin{pmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{pmatrix}$$

Estimating F

- *When solving matrix equations, one needs to take care of the conditioning of matrices. In the computation of Fundamental matrix, issues on conditioning needs to be taken care off.*
- We will go through the material below which is taken from the paper by Richard Hartley:
 - *"In Defence of the Eight-Point Algorithm" , IEEE Trans on Pattern Analysis and Machine Intelligence, vol. 19, no. 6, June 1997*

Estimating F (cont.)

\tilde{p}_1 and \tilde{p}_2 represent the image point on image 1 and image 2 respectively:

$$\tilde{p}_1 = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$\tilde{p}_2 = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

Given a set of point correspondences, i.e. pairs of \tilde{p}_1 and \tilde{p}_2 , we want to compute F using

$$\tilde{p}_2^T F \tilde{p}_1 = 0$$

Estimating F (cont.)

We can write the following to represent

$$\tilde{p}_2^T F \tilde{p}_1 = 0$$

$$\begin{bmatrix} u' & v' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0$$


$$F$$

Since $[u' \ v' \ 1]$ and $[u \ v \ 1]$ are all known through point correspondences, F is the only unknown to be solved. We can write the above equation in $Ax = b$ form.

Estimating F (cont.)

$$\begin{bmatrix} uu' & vu' & u' & uv' & vv' & v' & u & v & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

Estimating F (cont.)

The solution for f is the least eigenvector of $A^T A$.

Once we get the solution for f , we can form the 3×3 matrix F .

But F should be rank 2. How to enforce the rank 2 condition ?
-- we can use SVD to enforce the rank 2 condition.

Estimating F (cont.)

Suppose

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

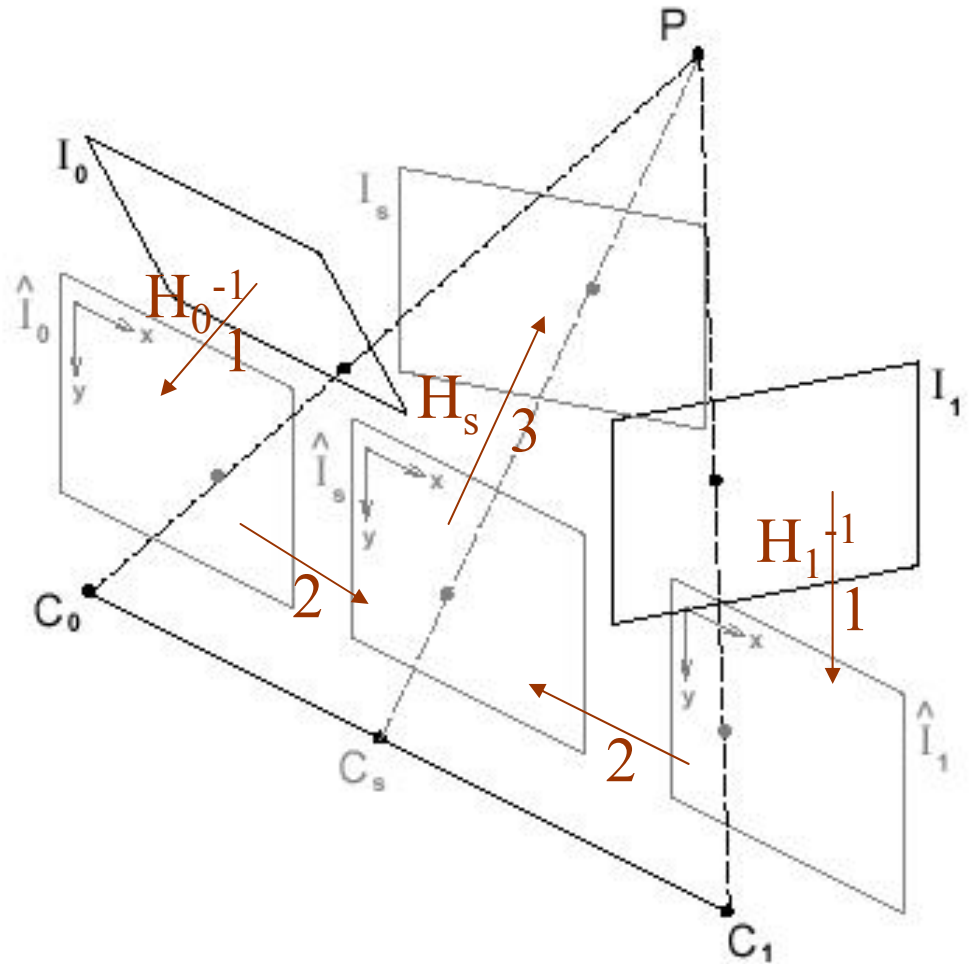
Let

$$F' = U \begin{bmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

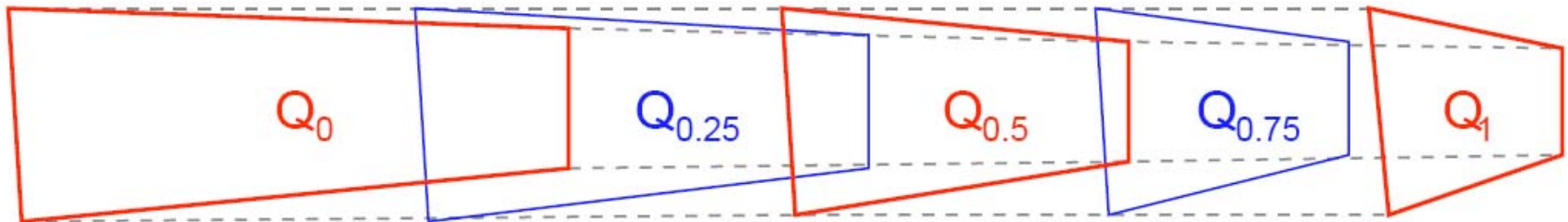
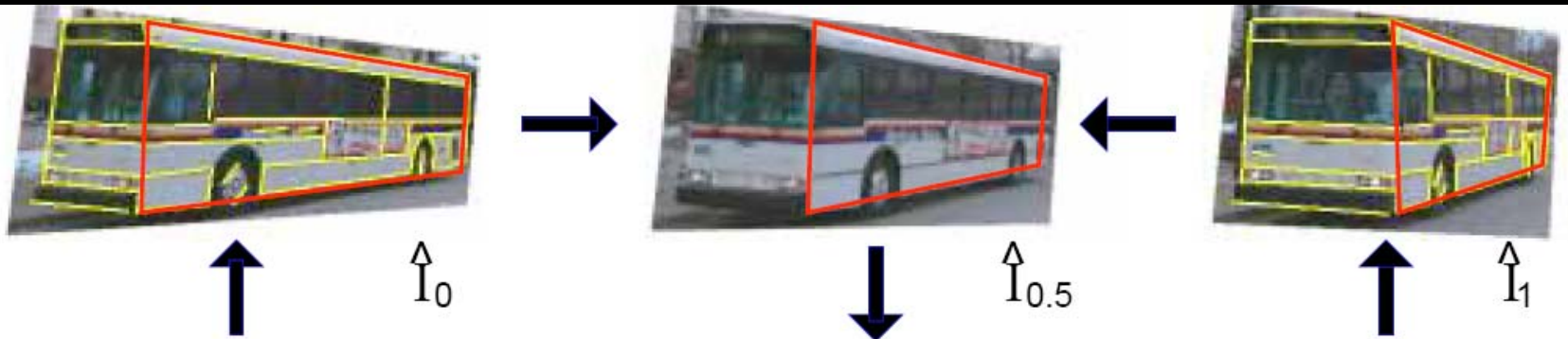
Then F' is the rank-2 matrix that most closely approximates F in the Frobenius norm sense i.e. $\|F - F'\|$ is minimum, where $\|\cdot\|$ denotes the Frobenius norm.

View Morphing

1. Pre-warp:
 - Warp the source and destination image to form parallel views.
2. Morphing:
 - Interpolating the prewarped images.
3. Post-warp
 - Warp the interpolated image to form the I_s



View Morphing



View Morphing

Step 1: Prewarp

$$\hat{I}_0 = H_0^{-1} I_0$$

$$\hat{I}_1 = H_1^{-1} I_1$$

Note that H_0 and H_1 are obtained by solving

$$H_1^T F H_0 = \hat{F}$$

where F is the fundamental matrix between I_0 and I_1

and

$$\hat{F} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

← parallel views

Step 2: Linearly interpolate positions and intensities of corresponding pixels in \hat{I}_0 and \hat{I}_1 to form

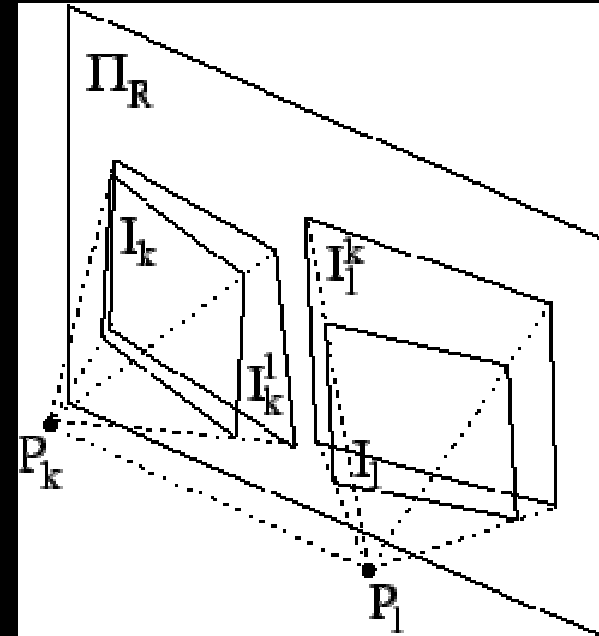
Step 3: Postwarp

$$I_s = H_s \hat{I}_s$$

Estimating the matrix H

- From non-parallel \rightarrow parallel.
 - The transformation is not unique.
 - Planar rectification in computer vision.
 - Rotate the epipole.

$$\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T = He, \text{ where } e \text{ is the epipole}$$



Estimating the matrix H (cont.)

■ Solutions suggested by the authors.

1. Rotate the epipole around an axis d_i .

$$R_{\theta_i}^{d_i}$$

- Make image planes parallel.

2. Rotate the image plane

$$R_{\phi_i}$$

- Make epipolar lines parallel.

3. Align the epipolar line T

$$W_0 = R_{\phi_0} R_{\theta_0}^{d_0}$$

$$W_1 = TR_{\phi_1} R_{\theta_1}^{d_1}$$

Estimating the matrix H (I)

■ Rotate around axis

$$d_0 = [d_0^x \quad d_0^y \quad 0]^T$$

$$e_0 = [e_0^x \quad e_0^y \quad e_0^z]^T$$

e_0 is the unit eigenvector of F corresponding to eigenvalue 0.

$$R_{\theta_0}^{d_0} = \begin{bmatrix} (d_0^x)^2 + (1 - (d_0^x)^2) \cos \theta_0 & d_0^x d_0^y (1 - \cos \theta_0) & d_0^y \sin \theta_0 \\ d_0^x d_0^y (1 - \cos \theta_0) & (d_0^y)^2 + (1 - (d_0^y)^2) \cos \theta_0 & -d_0^x \sin \theta_0 \\ -d_0^y \sin \theta_0 & d_0^x \sin \theta_0 & \cos \theta_0 \end{bmatrix}$$

$$\theta_0 = -\frac{\pi}{2} - \tan^{-1} \left(\frac{d_0^y e_0^x - d_0^x e_0^y}{e_0^z} \right)$$

Estimating the matrix H (II)

- The new epipoles are $\begin{bmatrix} \tilde{e}_i^x & \tilde{e}_i^y & 0 \end{bmatrix}^T = R_{\theta_i}^{d_i} e_i$
- Rotate the image plane so that epipolar lines are horizontal.

$$R_{\phi_i} = \begin{bmatrix} \cos \phi_i & -\sin \phi_i & 0 \\ \sin \phi_i & \cos \phi_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\phi_i = -\tan^{-1} \left(\frac{\tilde{e}_i^y}{\tilde{e}_i^x} \right)$$

Estimating the matrix H (III)

$$\begin{aligned}\tilde{F} &= \left(\left(R_{\phi_1} R_{\theta_1}^{d_1} \right)^{-1} \right)^T F \left(R_{\phi_0} R_{\theta_0}^{d_0} \right)^{-1} \\ &= R_{\phi_1} R_{\theta_1}^{d_1} F R_{\theta_0}^{-d_0} R_{-\phi_0} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & a \\ 0 & b & c \end{bmatrix}\end{aligned}$$

Our target \rightarrow

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

Multiply :

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -a & -c \\ 0 & 0 & b \end{bmatrix}$$

$$W_0 = R_{\phi_0} R_{\theta_0}^{d_0}$$

$$W_1 = T R_{\phi_1} R_{\theta_1}^{d_1}$$