



# Image Rectification (Stereo)

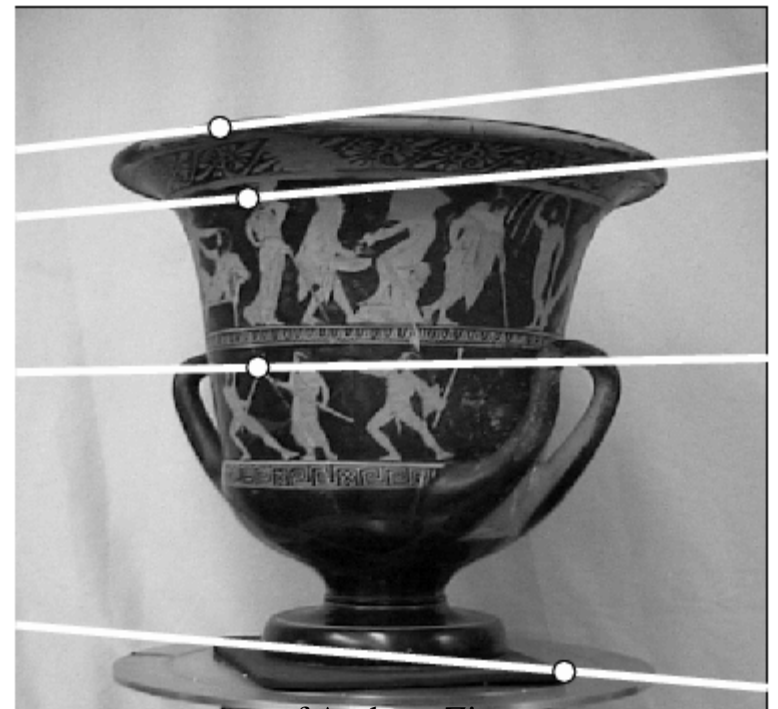
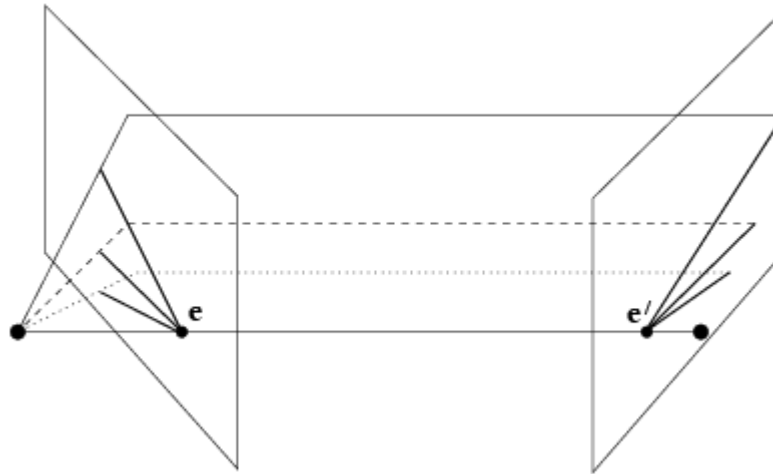
(New book: 7.2.1,  
old book: 11.1)

Guido Gerig  
CS 6320 Spring 2015

Credits: Prof. Mubarak Shah, Course notes modified from:  
<http://www.cs.ucf.edu/courses/cap6411/cap5415/>, Lecture 25



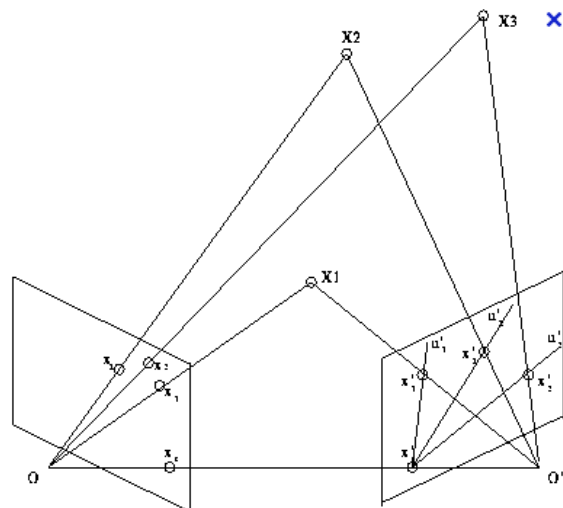
## Example: converging cameras



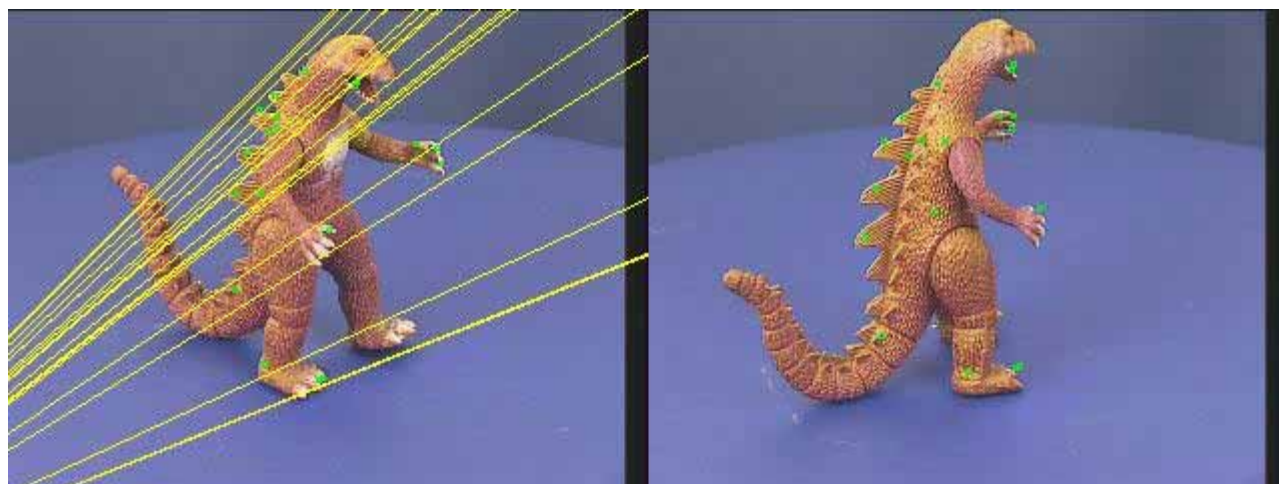
courtesy of Andrew Zisserman



# Epipolar Lines in Converging Cameras

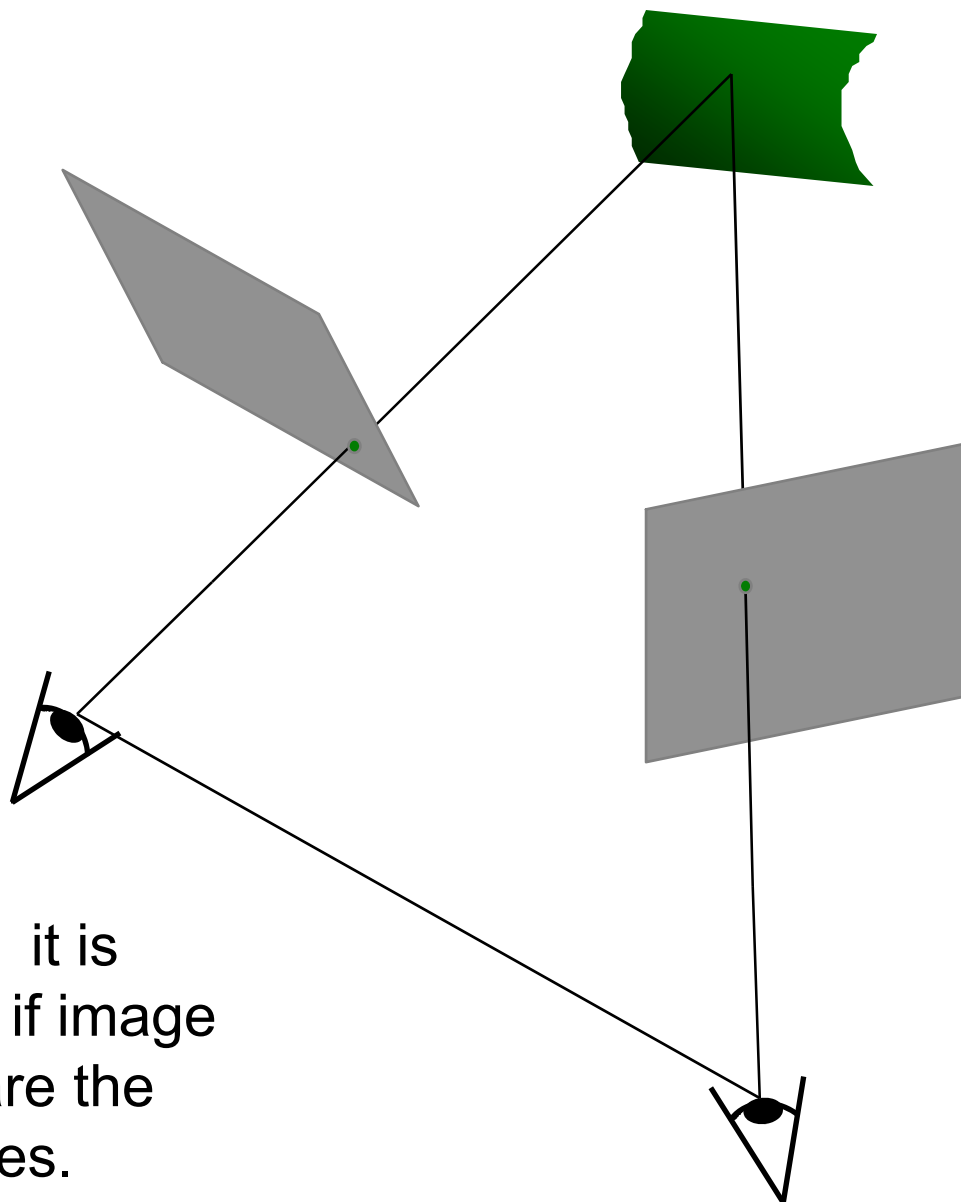


Epipolar lines all intersect at epipoles.





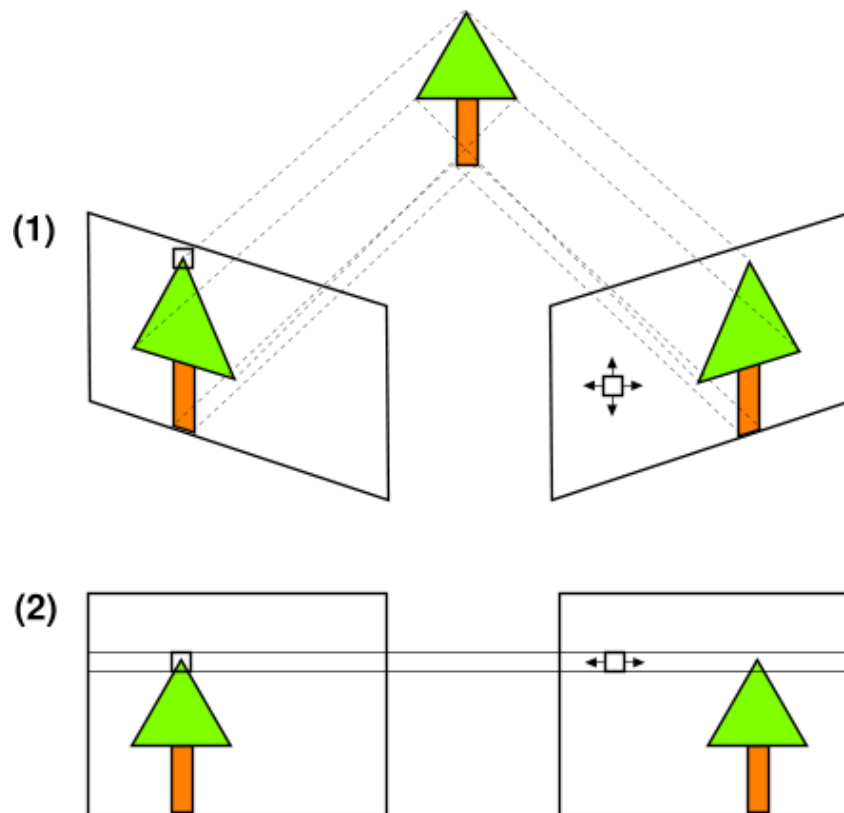
# Stereo image rectification



In practice, it is convenient if image scanlines are the epipolar lines.



# Image Rectification



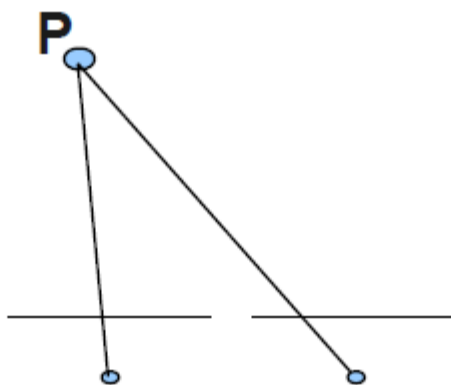
If the two cameras are aligned to be [coplanar](#), the search is simplified to one dimension - a horizontal line parallel to the [baseline](#) between the cameras

[http://en.wikipedia.org/wiki/Image\\_rectification](http://en.wikipedia.org/wiki/Image_rectification)



## Intuitions on Rectification

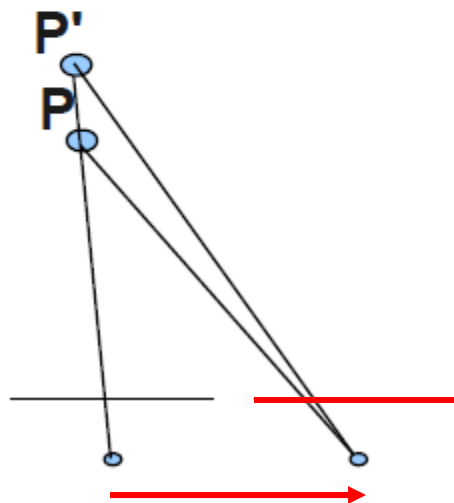
- Let's look in 2D
- If I change the *position* of my camera, then the location of the point depends on depth





## Intuitions on Rectification

- Let's look in 2D
- If I change the *position* of my camera, then the location of the point depends on depth

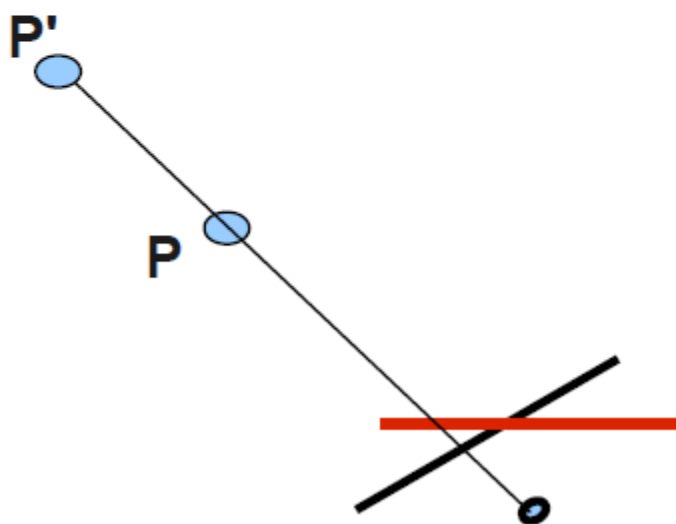


After translation, P and P' project onto separate locations.



## Intuitions on Rectification

- But if I just rotate the camera, then I don't need to know the depth
- Notice that  $P'$  and  $P$  still project to the same point in the rotated camera



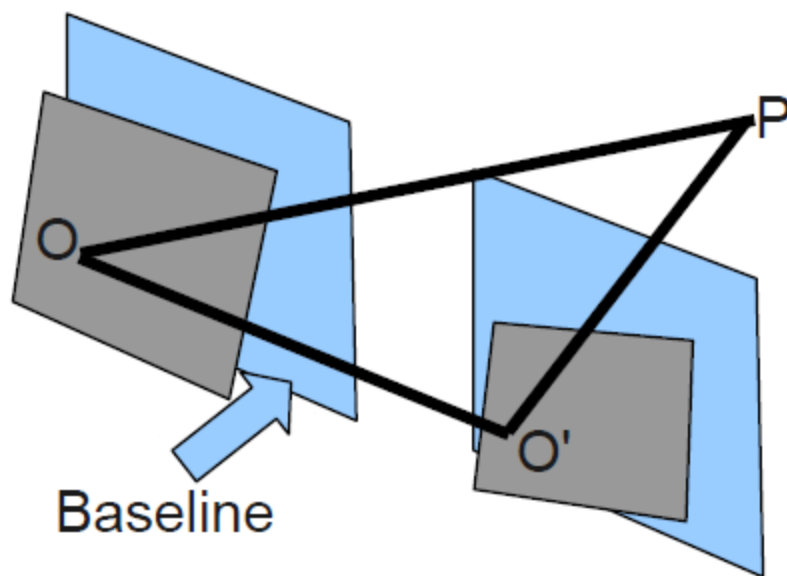
After rotation,  $P$  and  $P'$  still project onto the same location.





# Image Rectification

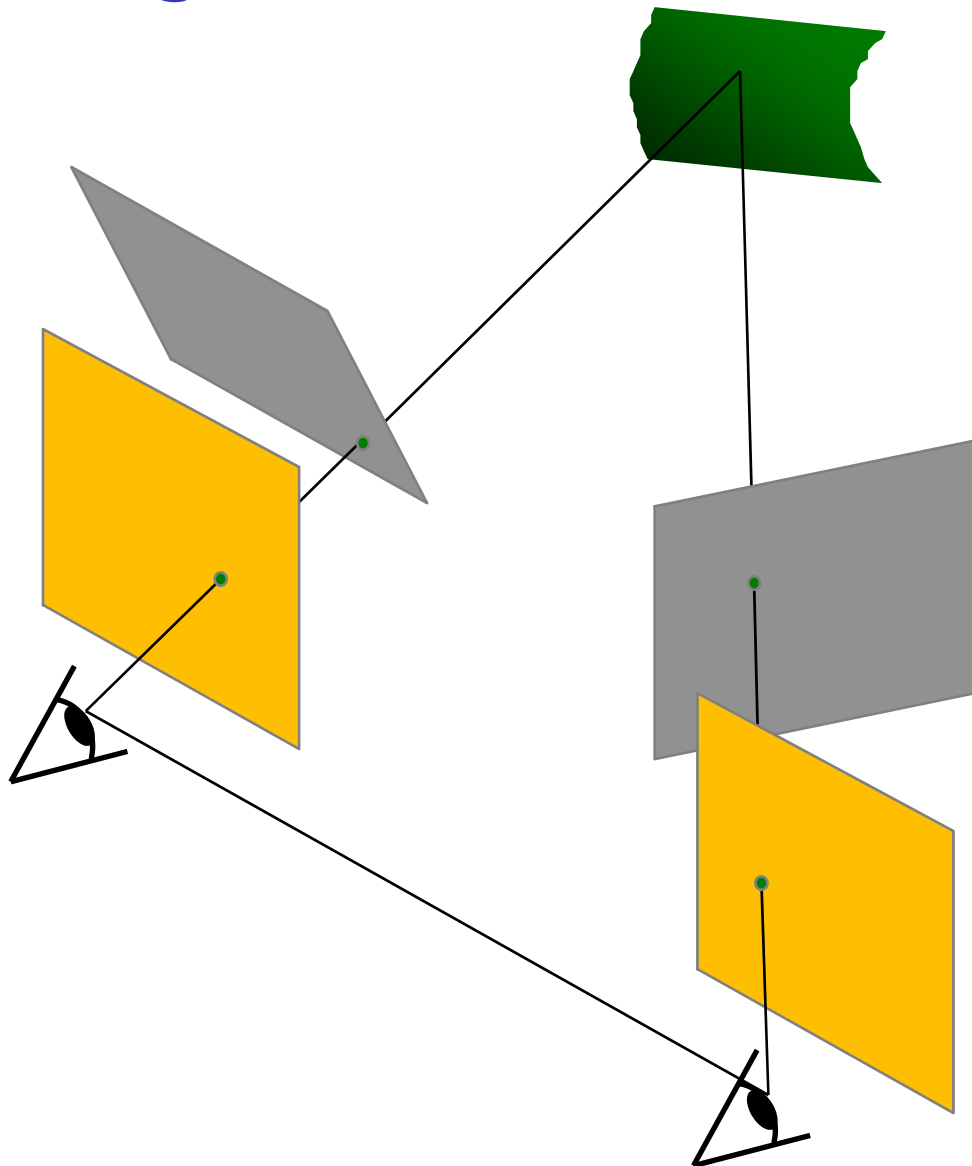
- I can rotate views to be parallel to each other and baseline





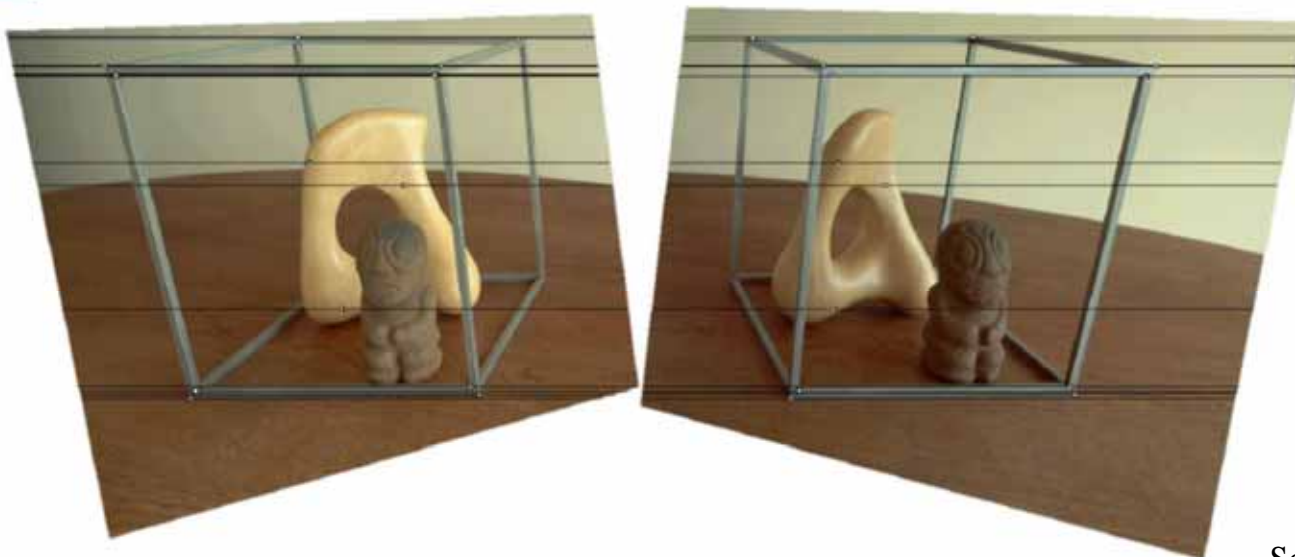
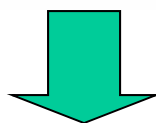
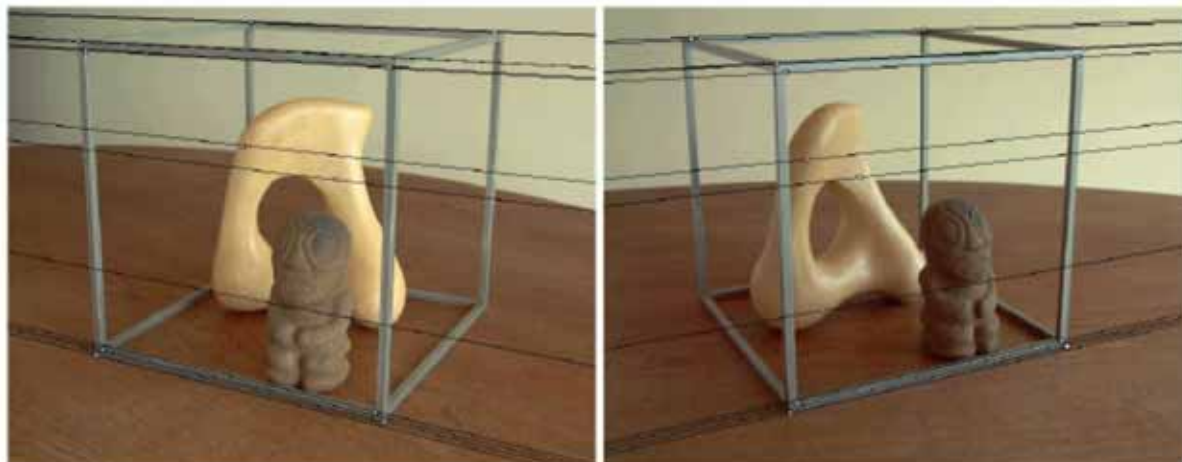
# Stereo image rectification

- Image Reprojection
  - reproject image planes onto common plane parallel to line between optical centers
  - a homography (3x3 transform) applied to both input images
  - pixel motion is horizontal after this transformation
  - C. Loop and Z. Zhang. [Computing Rectifying Homographies for Stereo Vision](#). IEEE Conf. Computer Vision and Pattern Recognition, 1999.





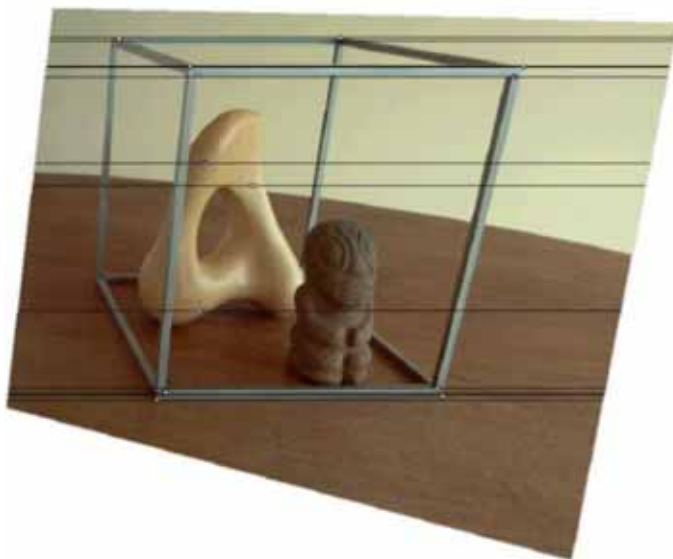
# Stereo image rectification:



Source: Alyosha Efros



# Stereo image rectification: example

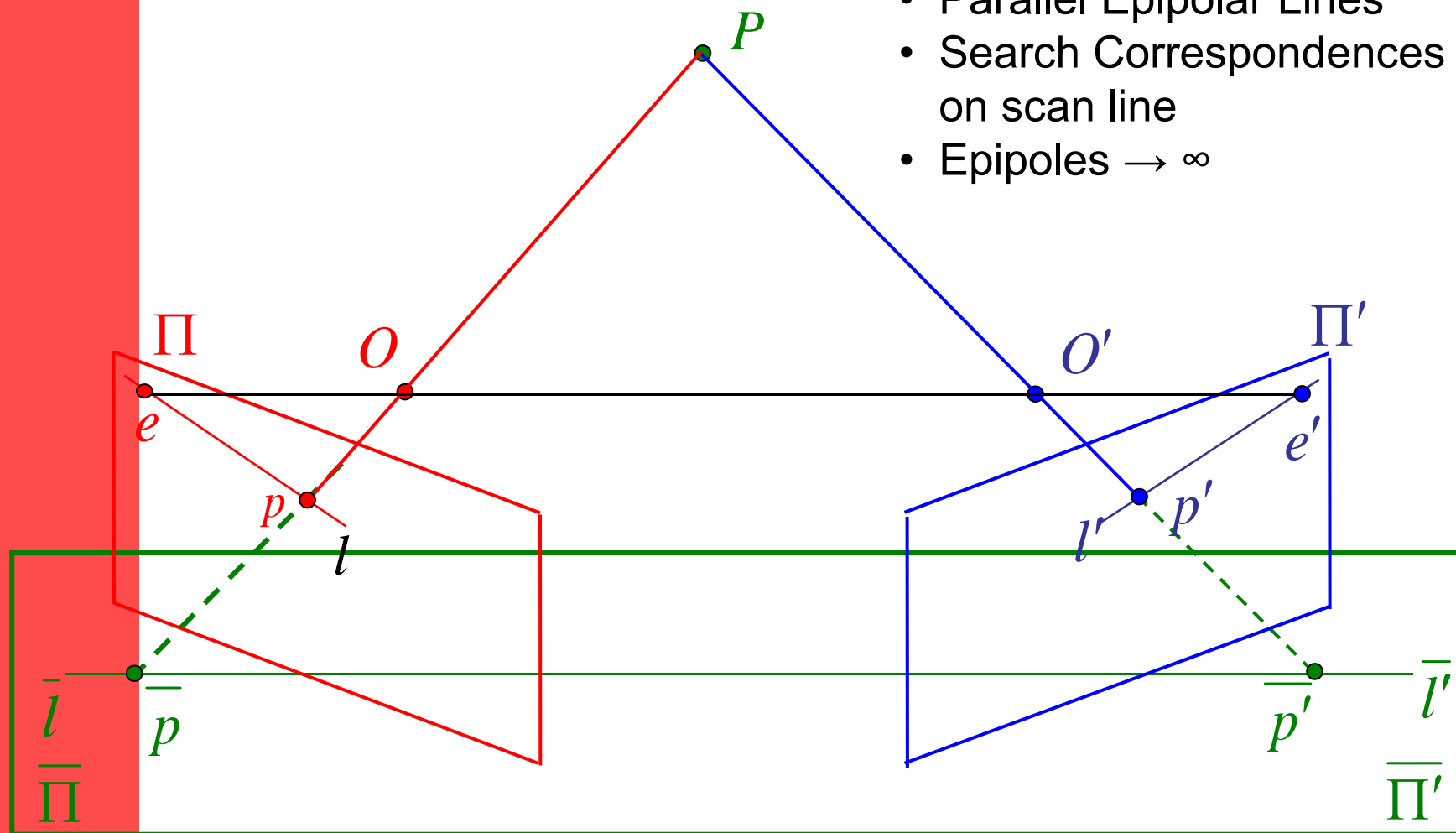


Source: Alyosha Efros



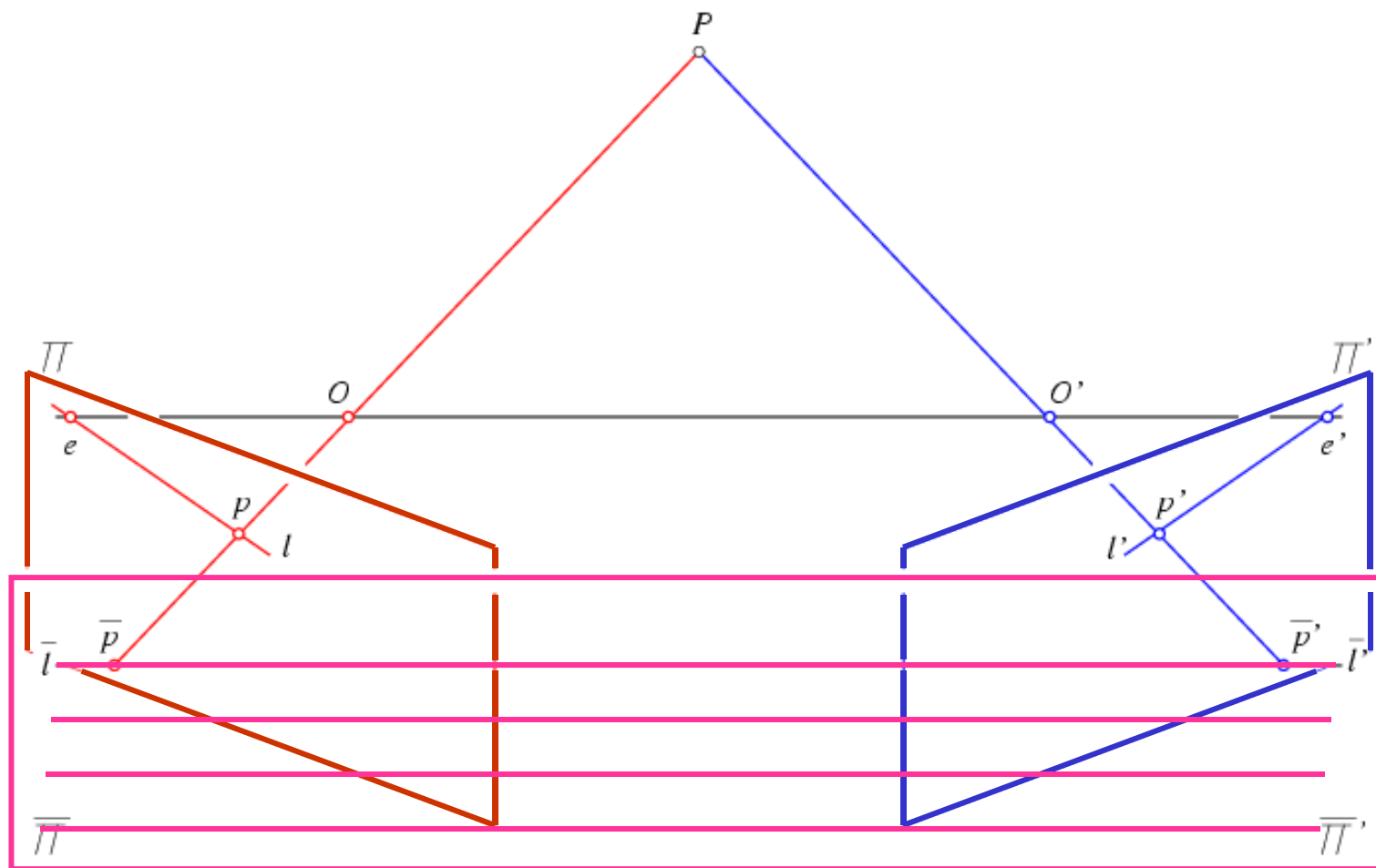
# Image Rectification

- Common Image Plane
- Parallel Epipolar Lines
- Search Correspondences on scan line
- Epipoles  $\rightarrow \infty$





# Image Rectification



All epipolar lines are parallel in the rectified image plane.



# Image Rectification

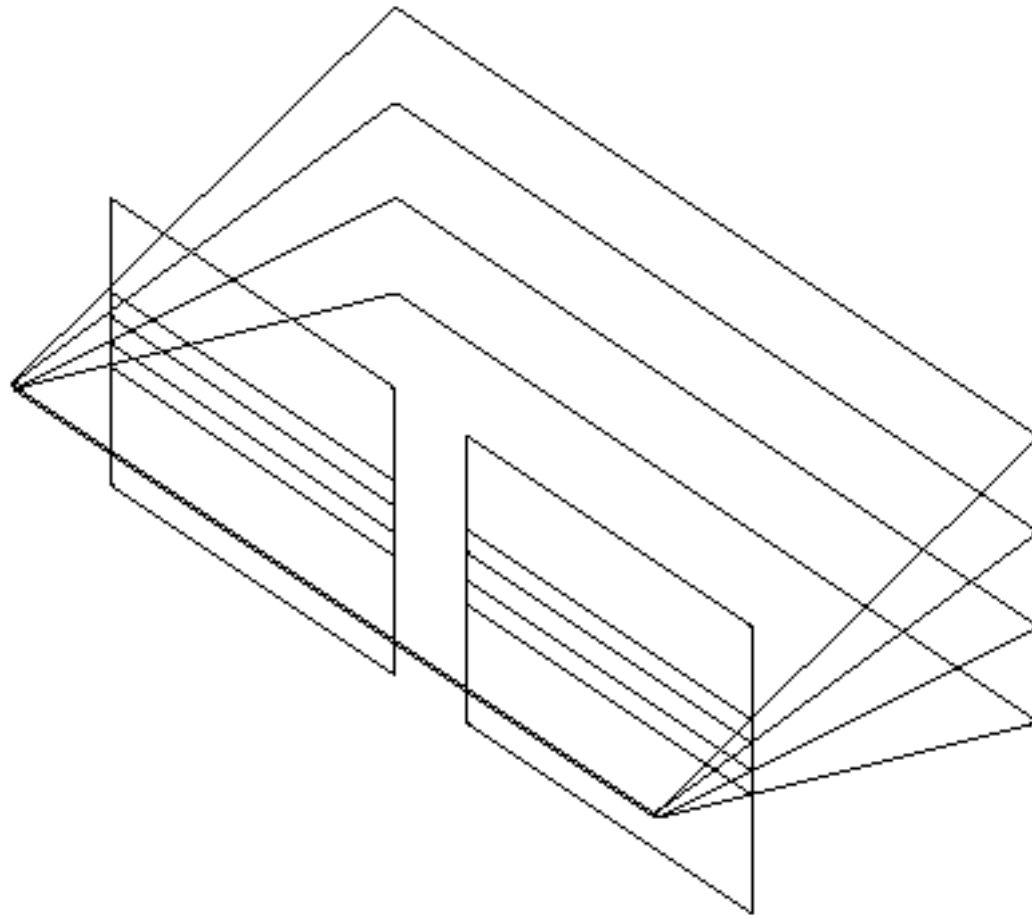


Figure 7.15: Standard stereo setup



# Algorithm Rectification

Following Trucco & Verri book pp. 159

- known  $T$  and  $R$  between cameras
- Rotate left camera so that epipole  $e_l$  goes to infinity along horizontal axis
- Apply same rotation to right camera to recover geometry
- Rotate right camera by  $R^{-1}$
- Adjust scale





The algorithm consists of four steps:

- Rotate the left camera so that the epipole goes to infinity along the horizontal axis.
- Apply the same rotation to the right camera to recover the original geometry.
- Rotate the right camera by  $R$ .
- Adjust the scale in both camera reference frames.

To carry out this method, we construct a triple of mutually orthogonal unit vectors  $\mathbf{e}_1$ ,  $\mathbf{e}_2$ , and  $\mathbf{e}_3$ . Since the problem is underconstrained, we are going to make an arbitrary choice. The first vector,  $\mathbf{e}_1$ , is given by the epipole; since the image center is in the origin,  $\mathbf{e}_1$  coincides with the direction of translation, or

$$\mathbf{e}_1 = \frac{\mathbf{T}}{\|\mathbf{T}\|}.$$

The only constraint we have on the second vector,  $\mathbf{e}_2$ , is that it must be orthogonal to  $\mathbf{e}_1$ . To this purpose, we compute and normalize the cross product of  $\mathbf{e}_1$  with the direction vector of the optical axis, to obtain

$$\mathbf{e}_2 = \frac{1}{\sqrt{T_x^2 + T_y^2}} [-T_y, T_x, 0]^\top.$$

The third unit vector is unambiguously determined as

$$\mathbf{e}_3 = \mathbf{e}_1 \times \mathbf{e}_2.$$

It is easy to check that the orthogonal matrix defined as

$$R_{rect} = \begin{pmatrix} \mathbf{e}_1^\top \\ \mathbf{e}_2^\top \\ \mathbf{e}_3^\top \end{pmatrix} \quad (7.22)$$

rotates the left camera about the projection center in such a way that the epipolar lines become parallel to the horizontal axis. This implements the first step of the algorithm. Since the remaining steps are straightforward, we proceed to give the customary algorithm:



### Algorithm RECTIFICATION

The input is formed by the intrinsic and extrinsic parameters of a stereo system and a set of points in each camera to be rectified (which could be the whole images). In addition, Assumptions 1 and 2 above hold.

1. Build the matrix  $R_{rect}$  as in (7.22);
2. Set  $R_l = R_{rect}$  and  $R_r = R R_{rect}$ ;
3. For each left-camera point,  $\mathbf{p}_l = [x, y, f]^T$  compute

$$R_l \mathbf{p}_l = [x', y', z']$$

and the coordinates of the corresponding rectified point,  $\mathbf{p}'_l$ , as

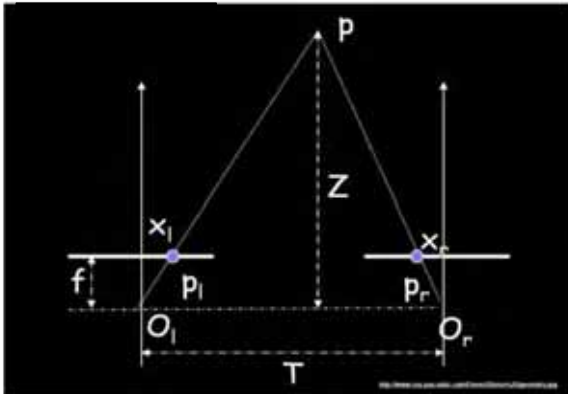
$$\mathbf{p}'_l = \frac{f}{z'} [x', y', z'].$$

4. Repeat the previous step for the right camera using  $R_r$  and  $\mathbf{p}_r$ .

The output is the pair of transformations to be applied to the two cameras in order to rectify the two input point sets, as well as the rectified sets of points.

From: Trucco & Verri, Introductory Techniques for 3-D Computer Vision, pp. 157-161

# Essential matrix example: parallel cameras



$$\mathbf{R} =$$

$$\mathbf{T} =$$

$$\mathbf{E} = [\mathbf{T}_x] \mathbf{R} =$$

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

For the parallel cameras,  
image of any point must lie on  
same horizontal line in each  
image plane.

Example I: compute the fundamental matrix for a parallel camera stereo rig

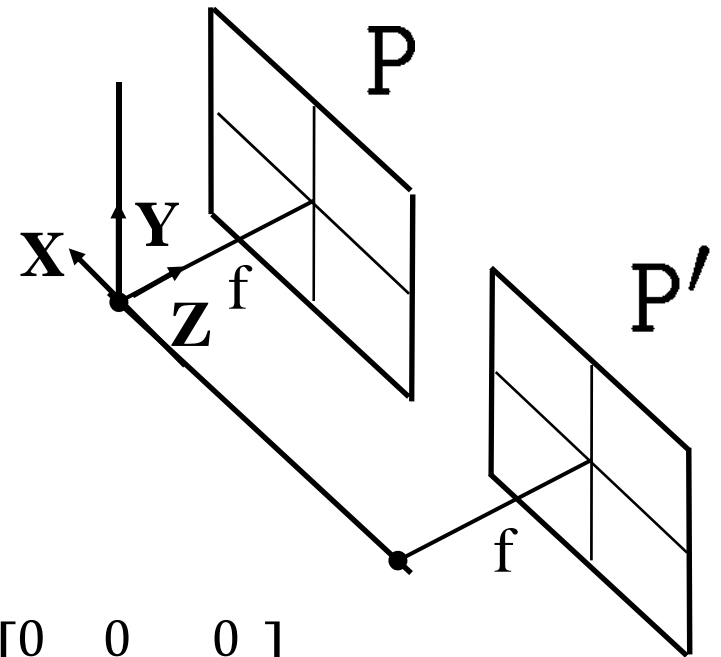
$$P = K[I \mid \mathbf{0}] \quad P' = K'[\mathbf{R} \mid \mathbf{t}]$$

$$K = K' = \begin{bmatrix} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R} = \mathbf{I} \quad \mathbf{t} = \begin{pmatrix} tx \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{F} = K'^{-\top} [\mathbf{t}]_{\times} \mathbf{R} K^{-1}$$

$$= \begin{bmatrix} \frac{1}{a} & 0 & 0 \\ 0 & \frac{1}{b} & 0 \\ -\frac{u_0}{a} & -\frac{v_0}{b} & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -tx \\ 0 & tx & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{a} & 0 & -\frac{u_0}{a} \\ 0 & \frac{1}{b} & -\frac{v_0}{b} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\frac{tx}{b} \\ 0 & \frac{tx}{b} & 0 \end{bmatrix}$$

$$\mathbf{x}'^{\top} \mathbf{F} \mathbf{x} = (x' \ y' \ 1) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$



- reduces to  $y = y'$  , i.e. raster correspondence (horizontal scan-lines)

F is a rank 2 matrix

The epipole  $e$  is the null-space vector (kernel) of F (exercise), i.e.  $Fe = 0$

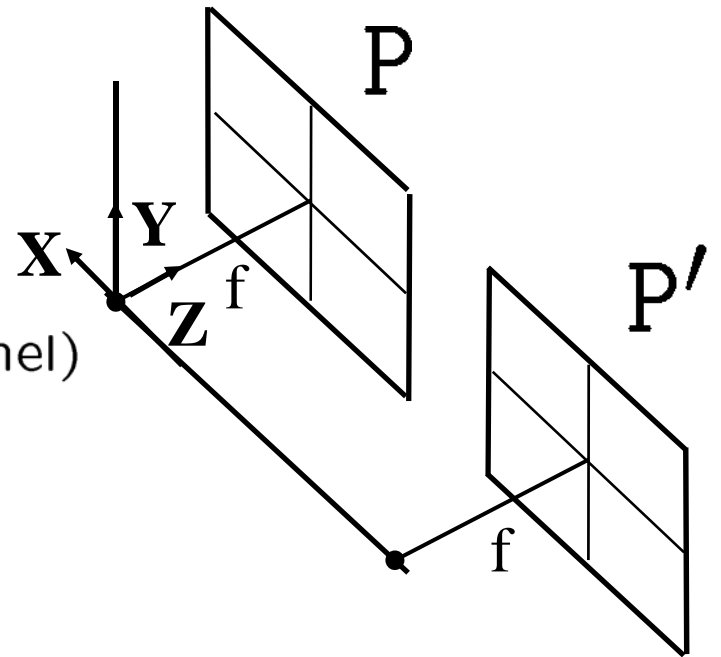
In this case

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

so that

$$e = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Geometric interpretation ?

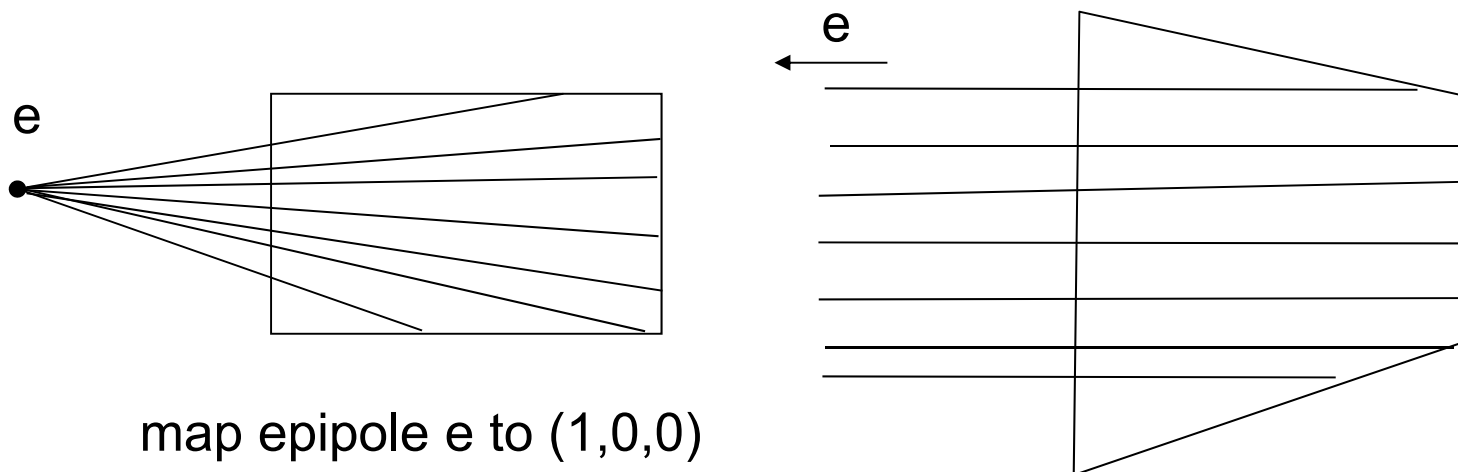




## Image pair rectification

Goal: Simplify stereo matching by “warping” the images

Apply projective transformation so that epipolar lines correspond to horizontal scanlines



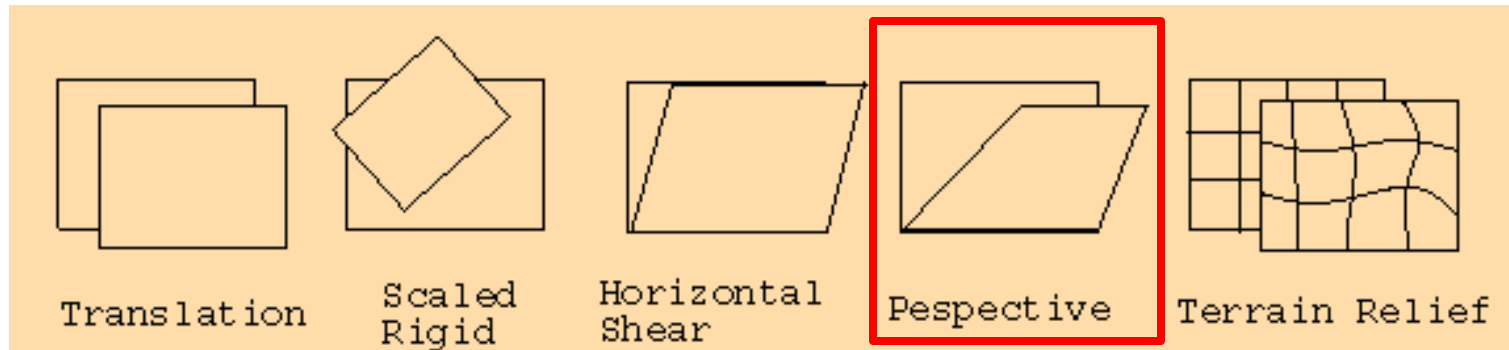
$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = H e$$

map epipole  $e$  to  $(1,0,0)$

try to minimize image distortion

problem when epipole in (or close to) the image

# Transformations



$$\begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

affine

$$\begin{pmatrix} X' \\ Y' \\ W \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

New image coordinates can be found as  $x' = X'/W$ ,  $y' = Y'/W$

$x'$ ,  $y'$ : homographies

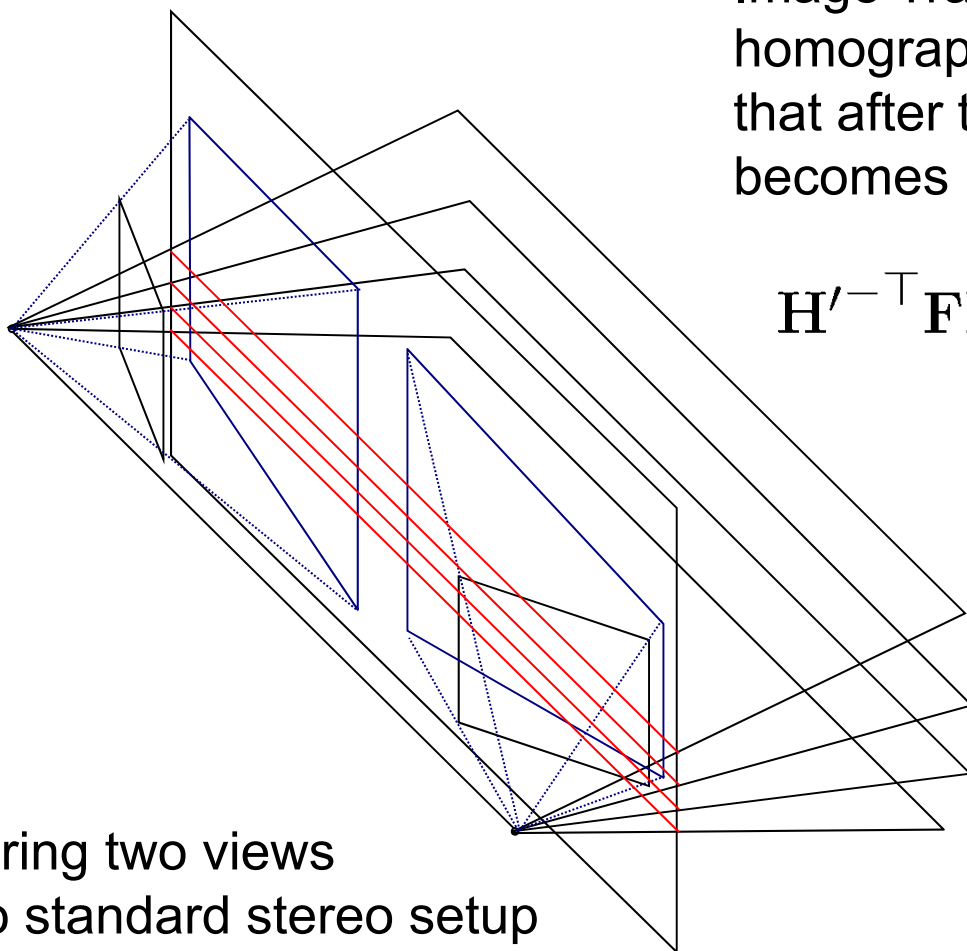
Homography: Projective Spaces



# Planar rectification

Image Transformations: Find homographies  $H'$  and  $H$  so that after transformations,  $F^{\text{new}}$  becomes  $F$  of parallel cameras:

$$H'^{-T} F H^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$



Bring two views  
to standard stereo setup

(moves epipole to  $\infty$ )

(not possible when in/close to image)





## More elegant Solution

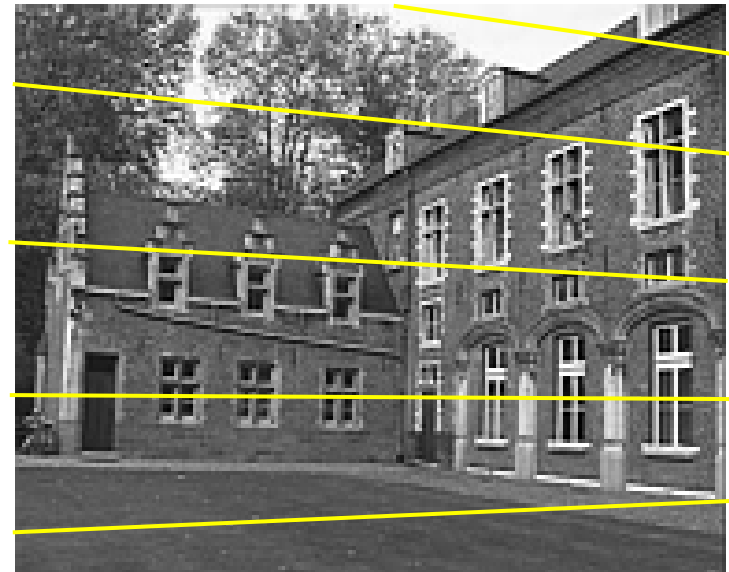
- Idea: Mapping epipole to infinity  $\rightarrow [1,0,0]^T$
- Factorization of matrix  $F=SM$ , where  $S$  is skew symmetric and  $M$  representing the required homography (projective transformation).
- Use SVD:

$$F = UDV^T = UWZD'V^T = (UWU^T) (UZD'V^T) = SM$$

where

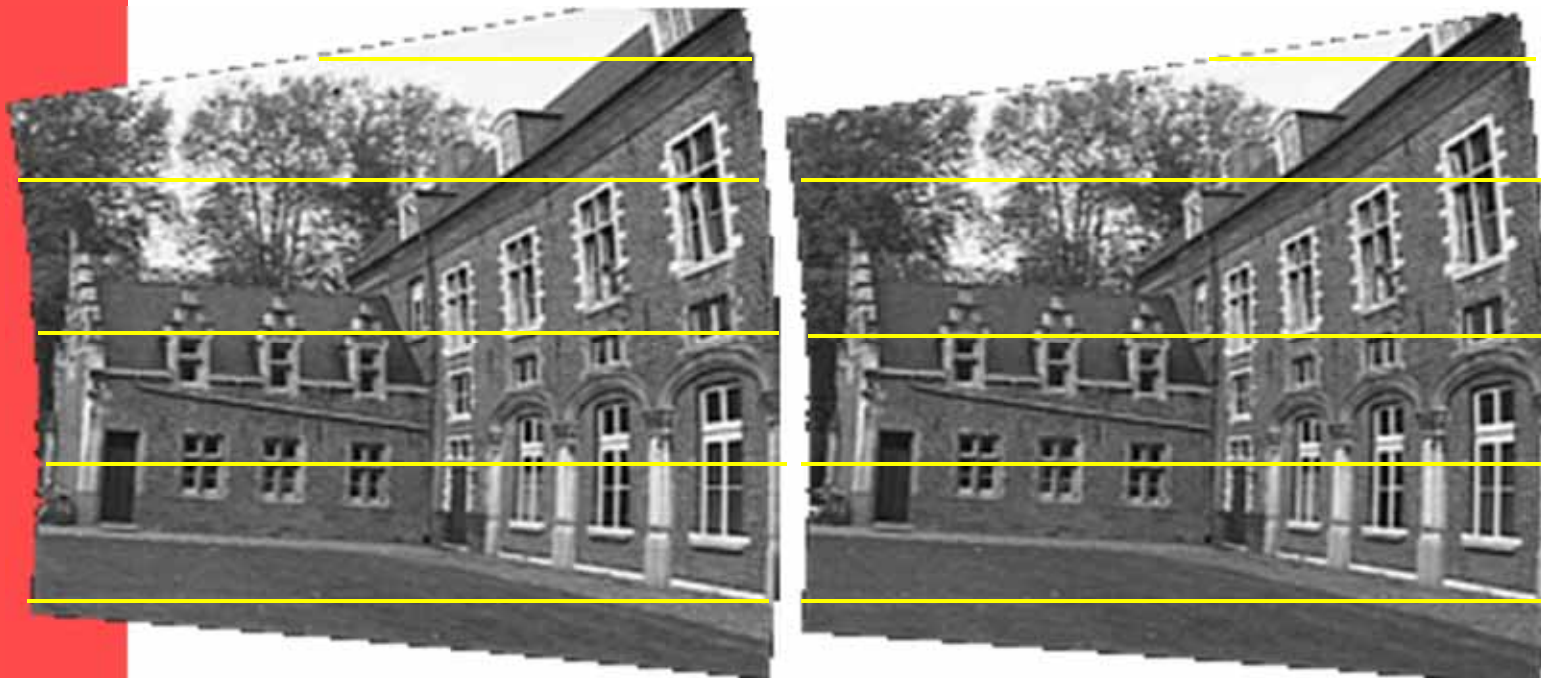
$$W = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} ; Z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

# Stereo matching with general camera configuration





# Image pair rectification

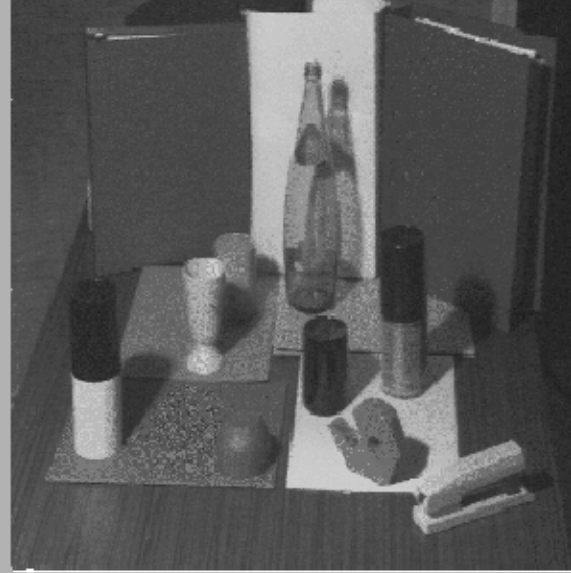




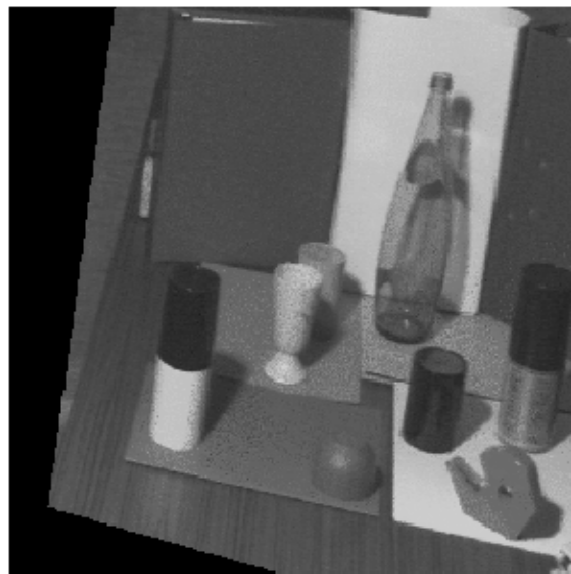
**Left image**



**Right image**

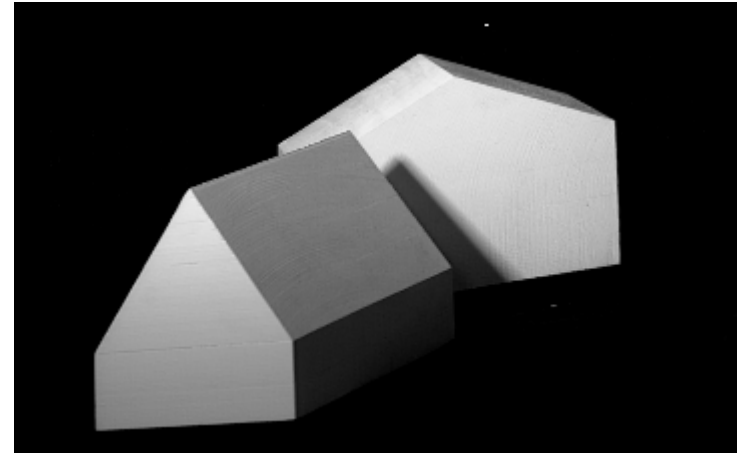
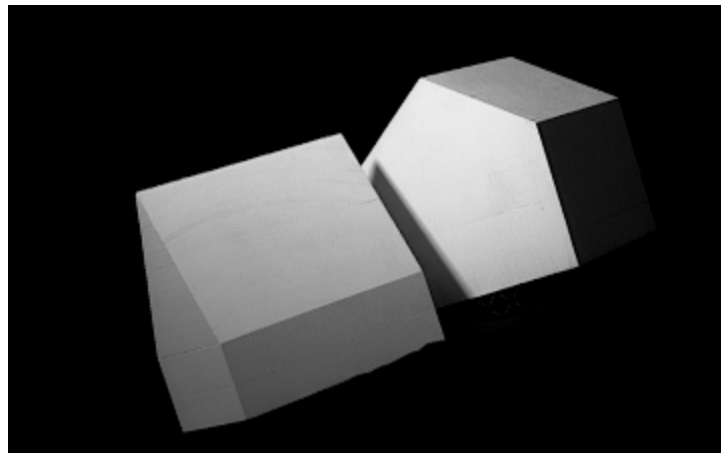
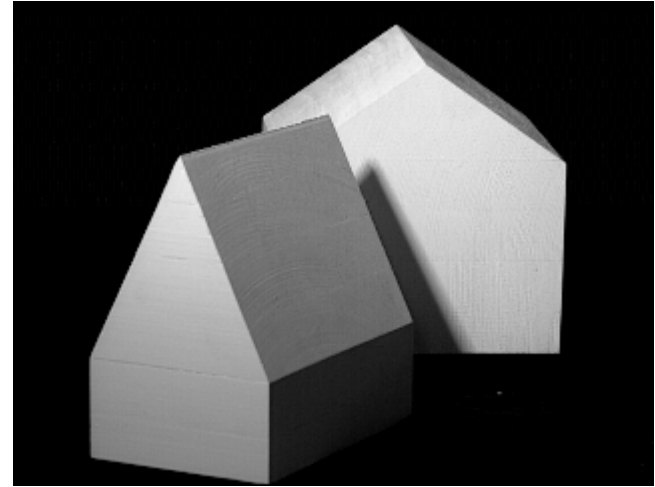
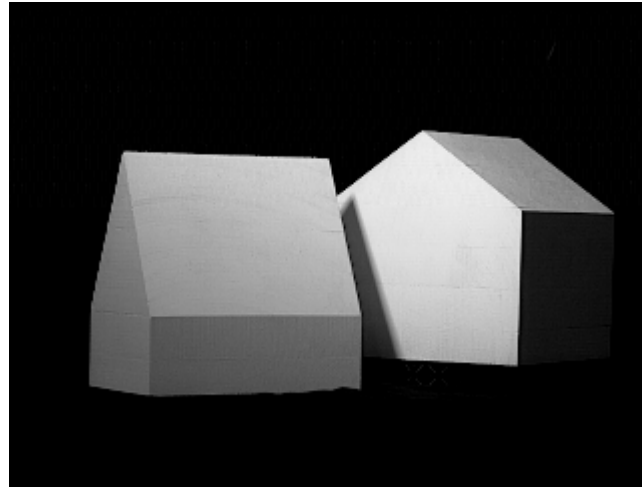


**Rectified left image**



**Rectified right image**











## Other Material /Code

- Epipolar Geometry, Rectification:  
[http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/FUSIELLO2/rectif\\_cvol.html](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FUSIELLO2/rectif_cvol.html)
- Fusiello, Trucco & Verri: Tutorial, Matlab code etc:  
<http://profs.sci.univr.it/~fusiello/demo/rect/>



# Run Example

## Updated Webpages:

<http://research.microsoft.com/en-us/um/people/zhang/INRIA/software.html>

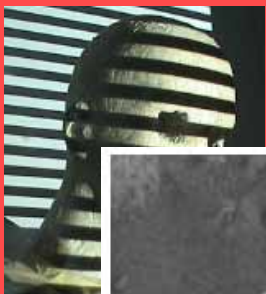
## SFM Example:

<http://research.microsoft.com/en-us/um/people/zhang/INRIA/SFM-Ex/SFM-Ex.html>

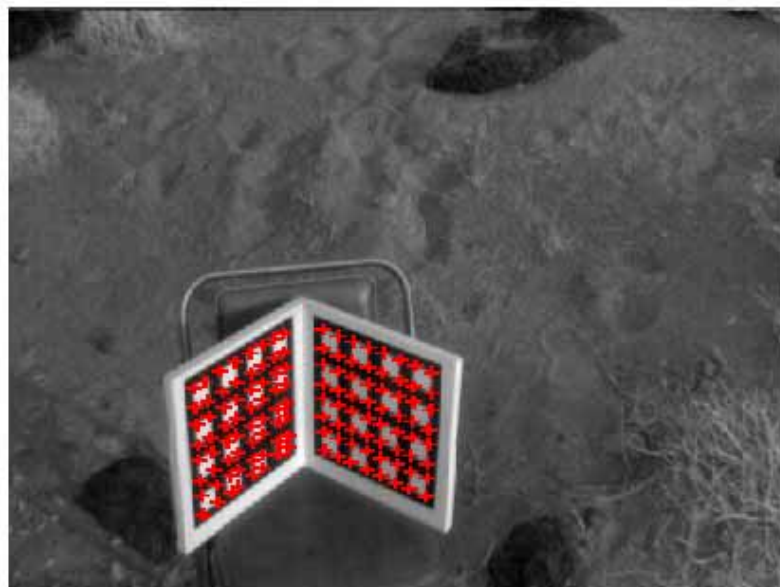
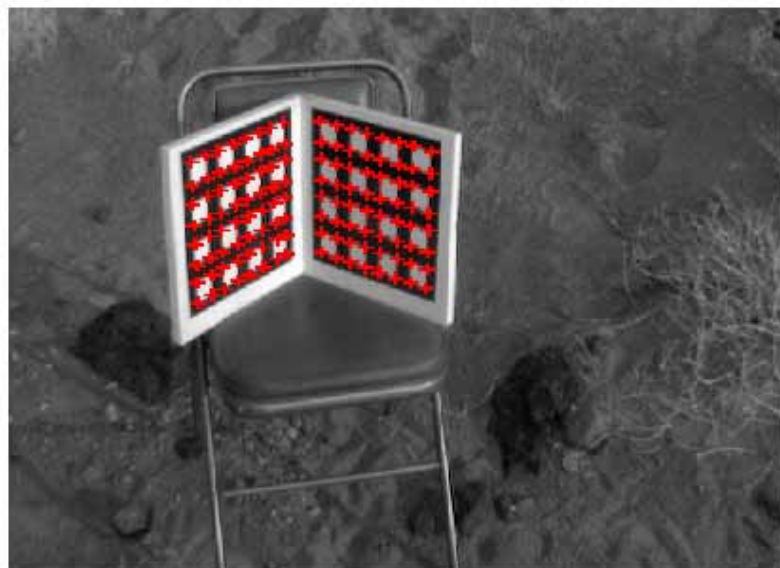
## Software:

<http://research.microsoft.com/en-us/um/people/zhang/INRIA/software.html>



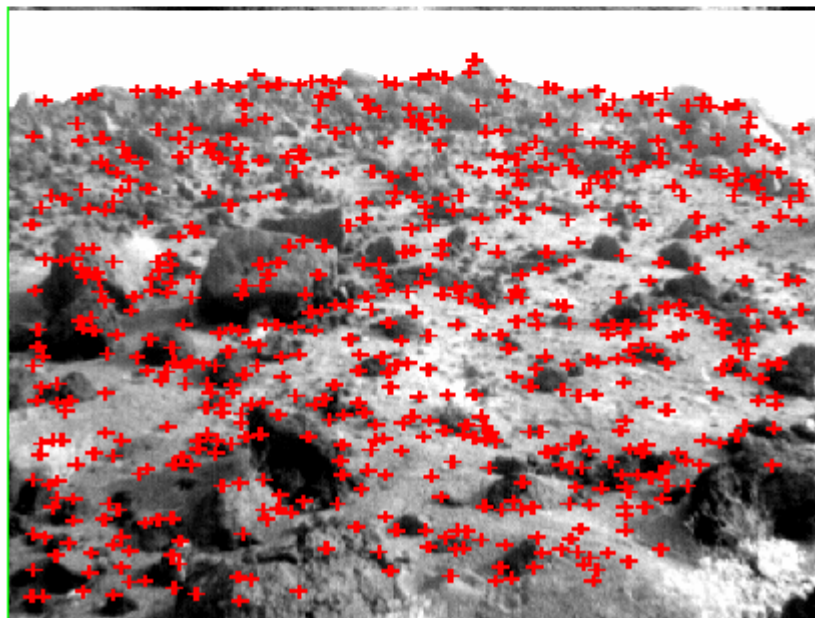
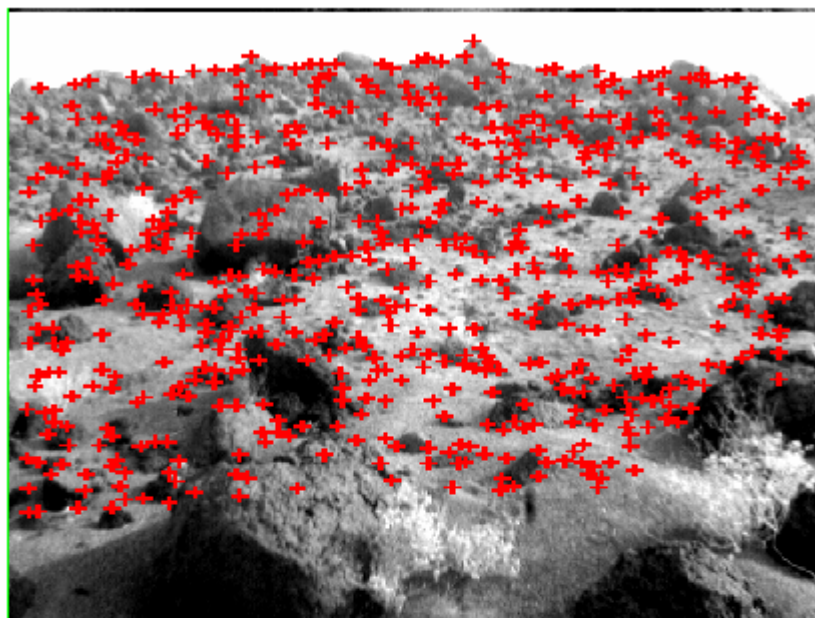


## Example: Zhengyou Zhang



Fundamental matrix between the two cameras:

5.2049303e-07	-2.9159992e-05	-2.5405448e-04
2.9090123e-05	6.7508671e-08	-5.6408271e-03
-5.0032429e-03	5.6687599e-03	1.0e-0

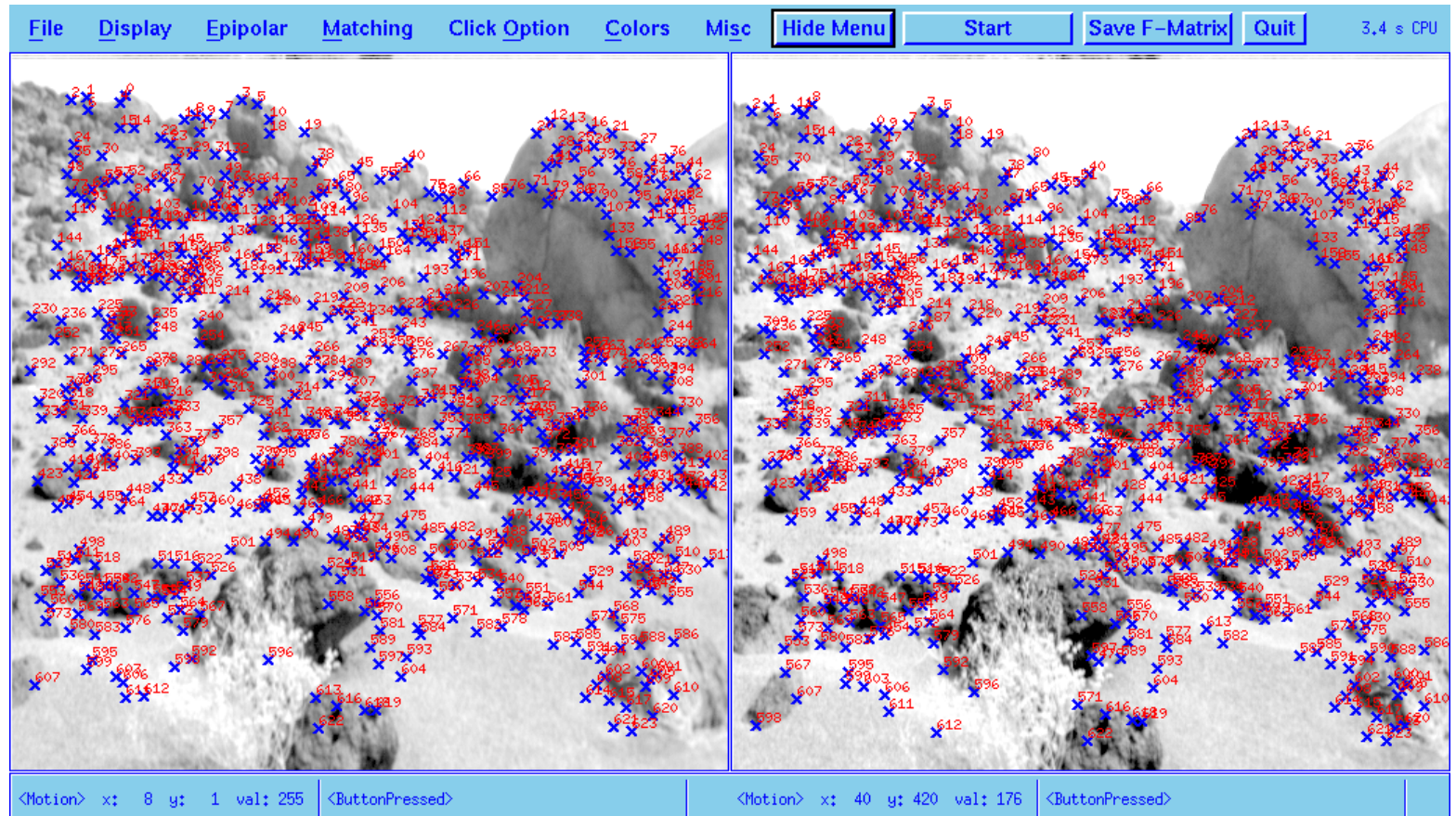


Points have been extracted using Harris corner detector, point matches via fundamental matrix  $F$  and search along epipolar lines.

2.019618e-06	1.887944e-04	-3.978027e-01
-1.702288e-04	-2.145494e-07	6.907326e-03
3.978027e-01	-7.553172e-03	8.266808e-01



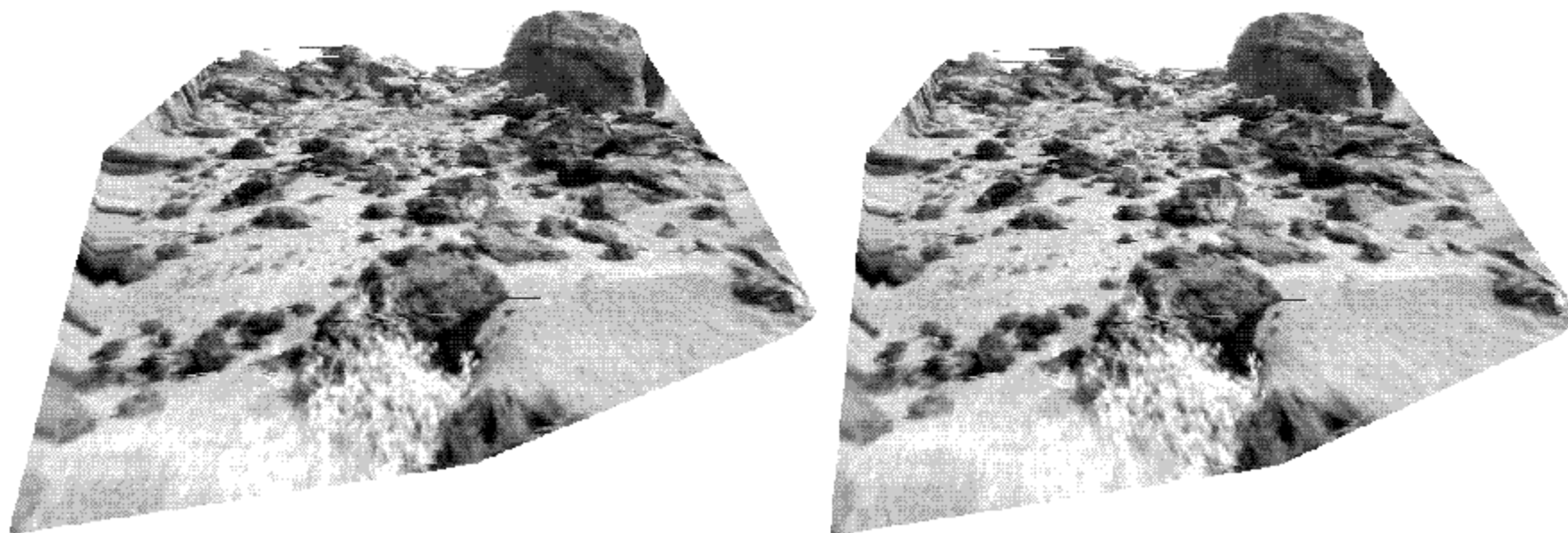
# Point matches found by a correlation technique







## 3D reconstruction represented by a pseudo stereogram



<http://research.microsoft.com/en-us/um/people/zhang/INRIA/SFM-Ex/SFM-Ex.html>