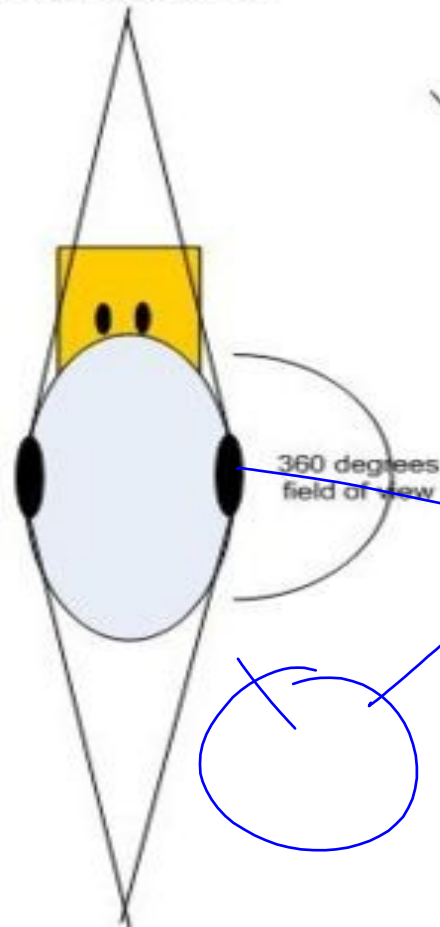


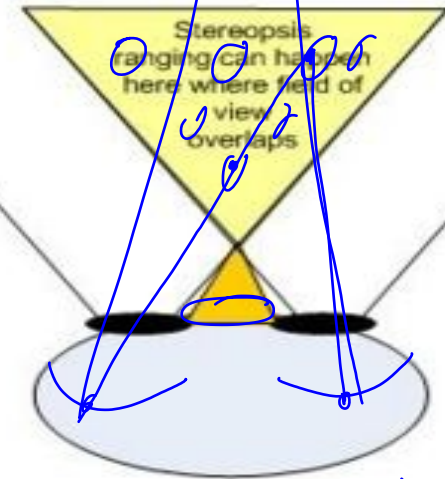
# Multiple Views

- Single image does not directly produce depth.
- Multiple images can produce depth by triangularization.

Duck, Rabbit, or Cow

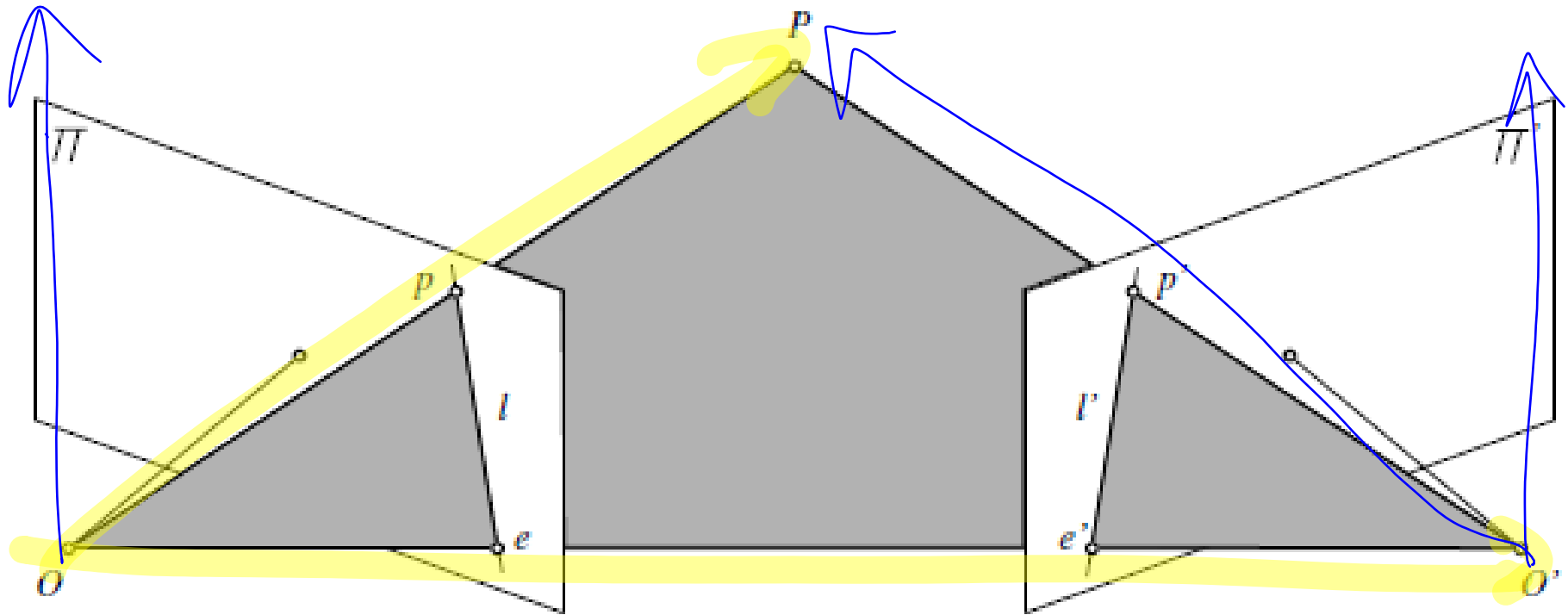


Owl, Cat or Person

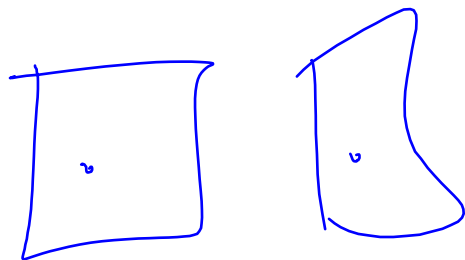


Shape from motion

# Epipolar Geometry



Epipolar plane, conjugated epipolar lines, epipoles, baseline



# Epipolar Constraint

$$P \in \mathcal{P}' = 0$$

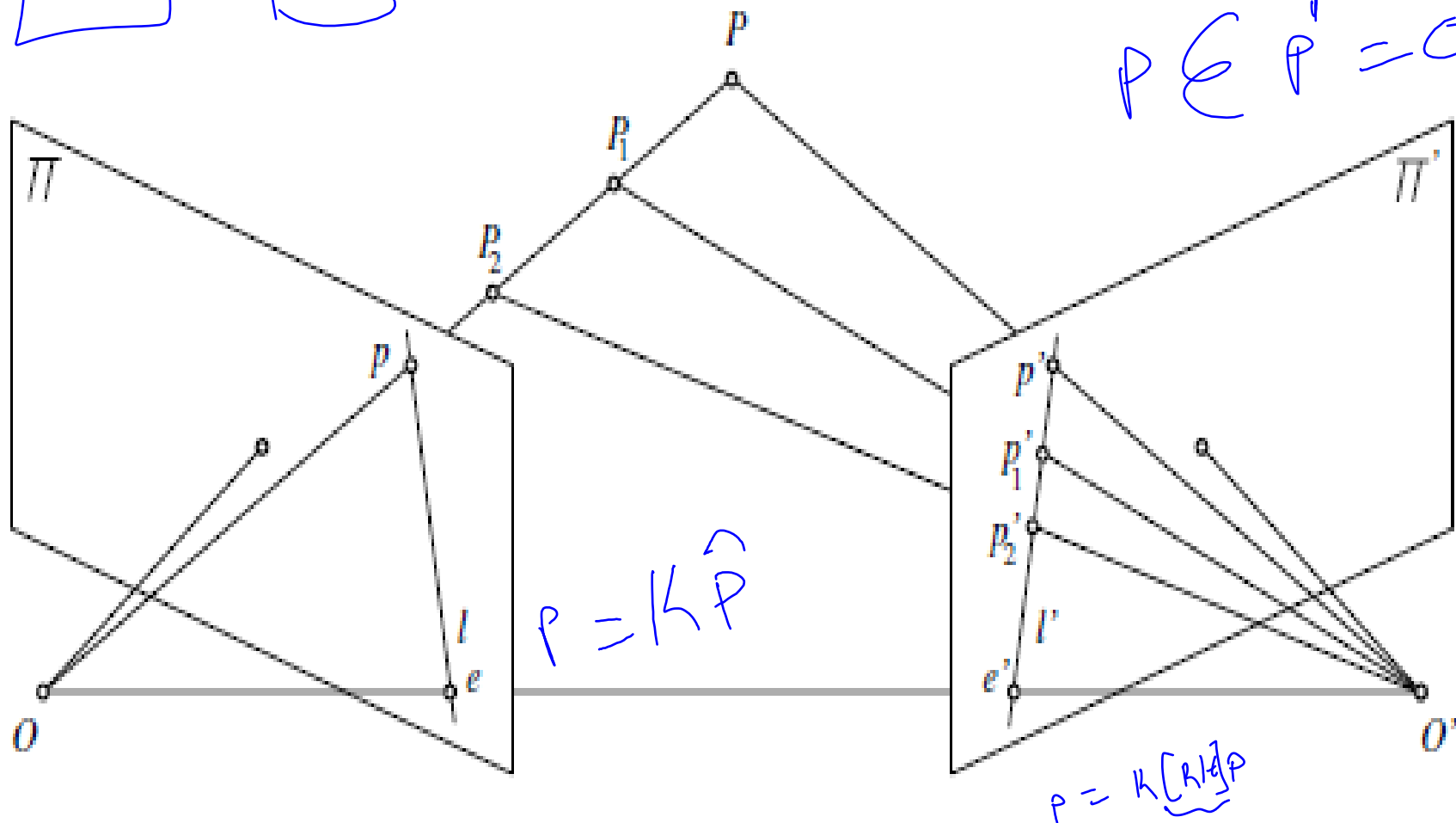


Figure 12.2. Epipolar constraint: given a calibrated stereo rig, the set of possible matches for the point  $p$  is constrained to lie on the associated epipolar line  $l'$ .

# Calibrated Epipolar Equations

Clearly, the epipolar constraint implies that the three vectors  $\vec{Op}$ ,  $\vec{O'p'}$ , and  $\vec{OO'}$  are coplanar. Equivalently, one of them must lie in the plane spanned by the other two, or

$$\vec{Op} \cdot [\vec{OO'} \times \vec{O'p'}] = 0.$$

We can rewrite this coordinate-independent equation in the coordinate frame associated to the first camera as

$$\mathbf{p} \cdot [\mathbf{t} \times (\mathbf{R}\mathbf{p}')],$$

where  $\mathbf{p}$  and  $\mathbf{p}'$  denote the homogenous image coordinate vectors,  $\mathbf{t}$  is the coordinate vector of the translation, and  $\mathbf{R}$  is the rotation matrix such that a free vector with coordinates  $\mathbf{w}$  in the second coordinate system has coordinates  $\mathbf{R}\mathbf{w}$  in the first one.

# Cross product in matrix form

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix}.$$

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} a_2 & a_3 \\ b_2 & b_3 \end{vmatrix} \mathbf{i} - \begin{vmatrix} a_1 & a_3 \\ b_1 & b_3 \end{vmatrix} \mathbf{j} + \begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix} \mathbf{k}$$

skew  
symmetric  
matrix

The vector cross product also can be expressed as the product of a skew-symmetric matrix and a vector:

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_{\times} \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad [\mathbf{a}]_{\times} \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

# Calibrated Epipolar Equations

$$\vec{O_p} \cdot [\vec{OO'} \times \vec{O'p'}] = 0. \quad \mathbf{p} \cdot [\mathbf{t} \times (\mathcal{R}\mathbf{p}')],$$

The above equations can be rewritten as

$$\mathbf{p}^T \mathcal{E} \mathbf{p}' = 0, \quad \text{where } \mathcal{E} = [\mathbf{t}_\times] \mathcal{R}$$

where  $\mathcal{E} = [\mathbf{t}_\times] \mathcal{R}$ , and  $[\mathbf{a}_\times]$  denotes the skew-symmetric matrix

# Essential Matrix

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

- The matrix E is called the essential matrix.
- Its nine coefficients are only defined up to scale,
- They can be parameterized by the three degrees of freedom of the rotation matrix R and the two degrees of freedom defining the direction of the translation vector t.
- $\mathbf{E} \mathbf{p}'$  can be interpreted as a line. ( $au + bv + c = 0$ )
- How do we write the other line equation? (take transpose)

# Fundamental Matrix

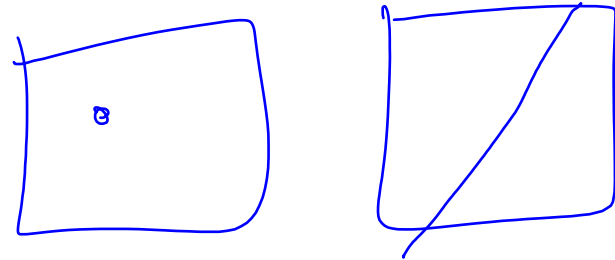
Essential matrix assumes the intrinsic parameters are known (calibrated case).

coordinates. When these parameters are unknown (*uncalibrated* cameras), we can write  $\mathbf{p} = \mathcal{K}\hat{\mathbf{p}}$  and  $\mathbf{p}' = \mathcal{K}'\hat{\mathbf{p}}'$ , where  $\mathcal{K}$  and  $\mathcal{K}'$  are  $3 \times 3$  calibration matrices, and  $\hat{\mathbf{p}}$  and  $\hat{\mathbf{p}}'$  are normalized image coordinate vectors. The Longuet-Higgins relation holds for these vectors, and we obtain

$$\mathbf{p}^T \mathcal{F} \mathbf{p}' = 0, \quad (12.1.4)$$

where the matrix  $\mathcal{F} = \mathcal{K}^{-T} \mathcal{E} \mathcal{K}'^{-1}$ , called the *fundamental matrix*, is not, in general,

- Fundamental Matrix is rank 2.
- It has 7 degrees of freedom.





# Weak Calibration

- How do we estimate the matrix  $F$  of stereo rig?

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

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

Take F33 as 1 and find 8 point correspondences.

$$\begin{pmatrix} u_1 u'_1 & u_1 v'_1 & u_1 & v_1 u'_1 & v_1 v'_1 & v_1 & u'_1 & v'_1 \\ u_2 u'_2 & u_2 v'_2 & u_2 & v_2 u'_2 & v_2 v'_2 & v_2 & u'_2 & v'_2 \\ u_3 u'_3 & u_3 v'_3 & u_3 & v_3 u'_3 & v_3 v'_3 & v_3 & u'_3 & v'_3 \\ u_4 u'_4 & u_4 v'_4 & u_4 & v_4 u'_4 & v_4 v'_4 & v_4 & u'_4 & v'_4 \\ u_5 u'_5 & u_5 v'_5 & u_5 & v_5 u'_5 & v_5 v'_5 & v_5 & u'_5 & v'_5 \\ u_6 u'_6 & u_6 v'_6 & u_6 & v_6 u'_6 & v_6 v'_6 & v_6 & u'_6 & v'_6 \\ u_7 u'_7 & u_7 v'_7 & u_7 & v_7 u'_7 & v_7 v'_7 & v_7 & u'_7 & v'_7 \\ u_8 u'_8 & u_8 v'_8 & u_8 & v_8 u'_8 & v_8 v'_8 & v_8 & u'_8 & v'_8 \end{pmatrix} \begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = - \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

# Stereopsis

This section is concerned with the design and implementation of algorithms that mimic our ability to perform estimation of depth from two images, known as stereopsis.

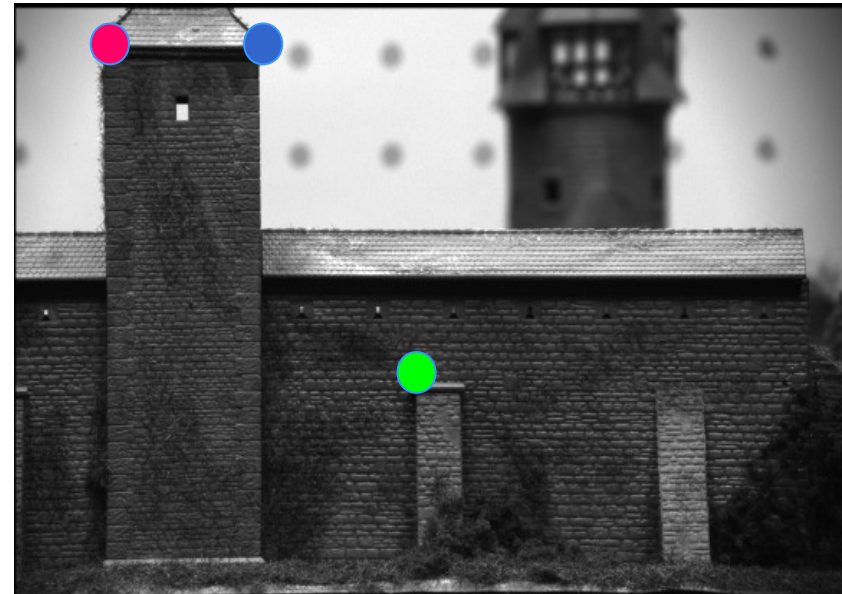
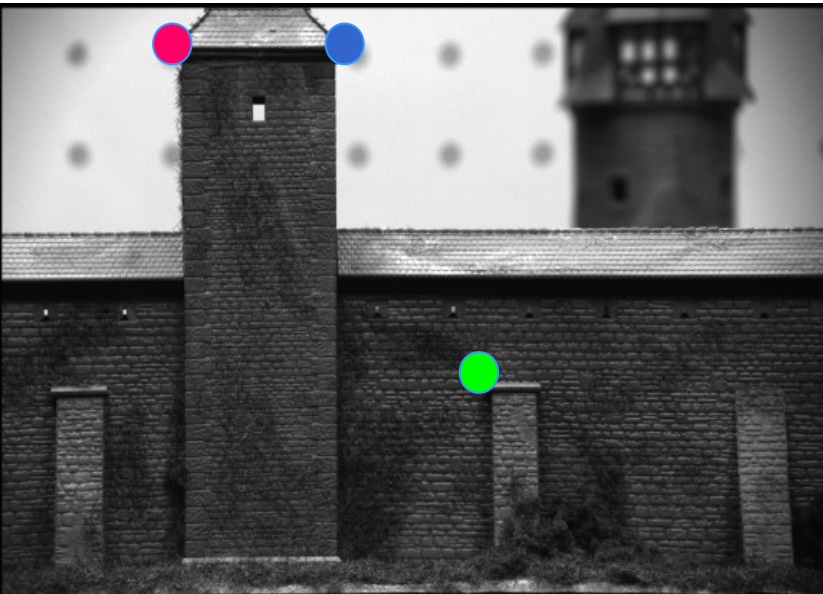
Stereo vision involves two processes:

- The binocular fusion of features observed by the two eyes
- the reconstruction of their three-dimensional preimage.

*stereo correspondence problem*

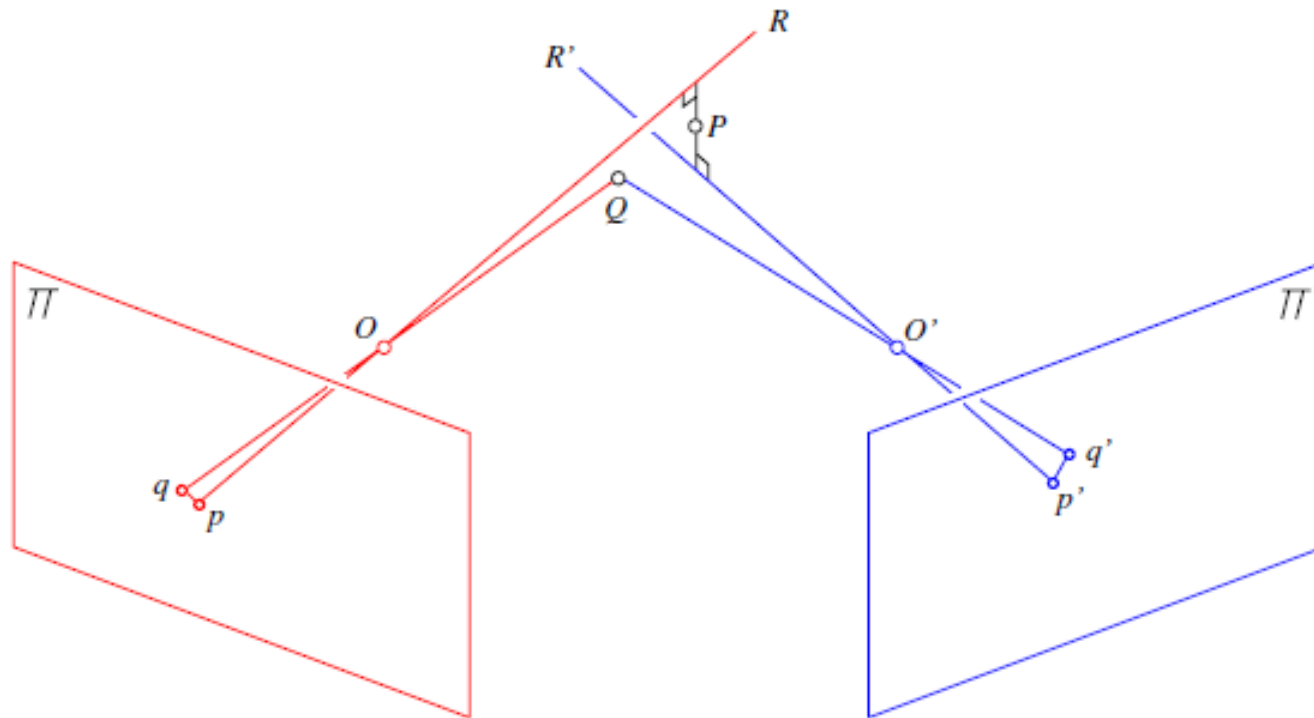


# Finding Correspondences:



# Reconstruction

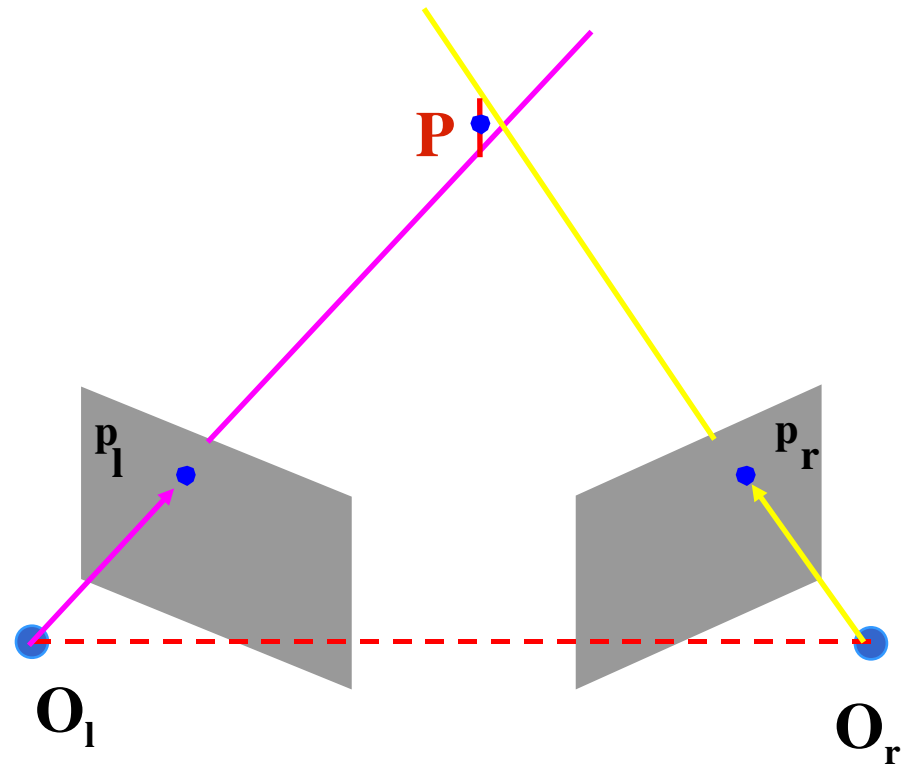
Given a calibrated stereo rig and two matching image points  $p$  and  $p'$ , it is in principle straightforward to reconstruct the corresponding scene point by intersecting the two rays  $R = Op$  and  $R' = O'p'$ . However, the rays  $R$  and  $R'$  will never, in practice, actually intersect, due to calibration and feature localization errors (Figure 13.4).



**Figure 13.4.** Triangulation in the presence of measurement errors. See text for details.

# Reconstruction by Triangulation

- Assumption and Problem
  - Under the assumption that both intrinsic and extrinsic parameters are known
  - Compute the 3-D location from their projections,  $p_l$  and  $p_r$
- Solution
  - **Triangulation:** Two rays are known and the intersection can be computed
  - Problem: **Two rays will not actually intersect in space due to errors in calibration and correspondences, and pixelization**
  - Solution: find a point in space with minimum distance from both rays





# Reconstruction by Triangulation

## Algorithm TRIANG

All vectors and coordinates are referred to the left camera reference frame. The input is formed by a set of corresponding points; let  $\mathbf{p}_l$  and  $\mathbf{p}_r$  be a generic pair.

Let  $a\mathbf{p}_l$ ,  $a \in \mathbb{R}$ , be the ray  $l$  through  $O_l$  ( $a = 0$ ) and  $\mathbf{p}_l$  ( $a = 1$ ). Let  $\mathbf{T} + bR^\top \mathbf{p}_r$ ,  $b \in \mathbb{R}$ , the ray  $r$  through  $O_r$  ( $b = 0$ ) and  $\mathbf{p}_r$  ( $b = 1$ ). Let  $\mathbf{w} = \mathbf{p}_l \times R^\top \mathbf{p}_r$  the vector orthogonal to both  $l$  and  $r$ , and  $a\mathbf{p}_l + c\mathbf{w}$ ,  $c \in \mathbb{R}$ , the line  $w$  through  $a\mathbf{p}_l$  (for some fixed  $a$ ) and parallel to  $\mathbf{w}$ .

1. Determine the endpoints of the segment,  $s$ , belonging to the line parallel to  $\mathbf{w}$  that joins  $l$  and  $r$ ,  $a_0\mathbf{p}_l$  and  $\mathbf{T} + b_0R^\top \mathbf{p}_r$ , by solving 
$$a\mathbf{p}_l - bR^\top \mathbf{p}_r + c(\mathbf{p}_l \times R^\top \mathbf{p}_r) = \mathbf{T}$$
2. The triangulated point,  $P'$ , is the midpoint of the segment  $s$ .

The output is the set of reconstructed 3-D points.



$$\boldsymbol{p} = \frac{1}{z} \boldsymbol{M} \boldsymbol{P}, \quad \text{where} \quad \boldsymbol{M} = \boldsymbol{\mathcal{K}}(\mathcal{R}, \boldsymbol{t}),$$

$$\boldsymbol{\mathcal{K}} \stackrel{\text{def}}{=} \begin{pmatrix} \alpha & -\alpha \cot \theta & u_0 \\ 0 & \frac{\beta}{\sin \theta} & v_0 \\ 0 & 0 & 1 \end{pmatrix}.$$

# Reconstruction

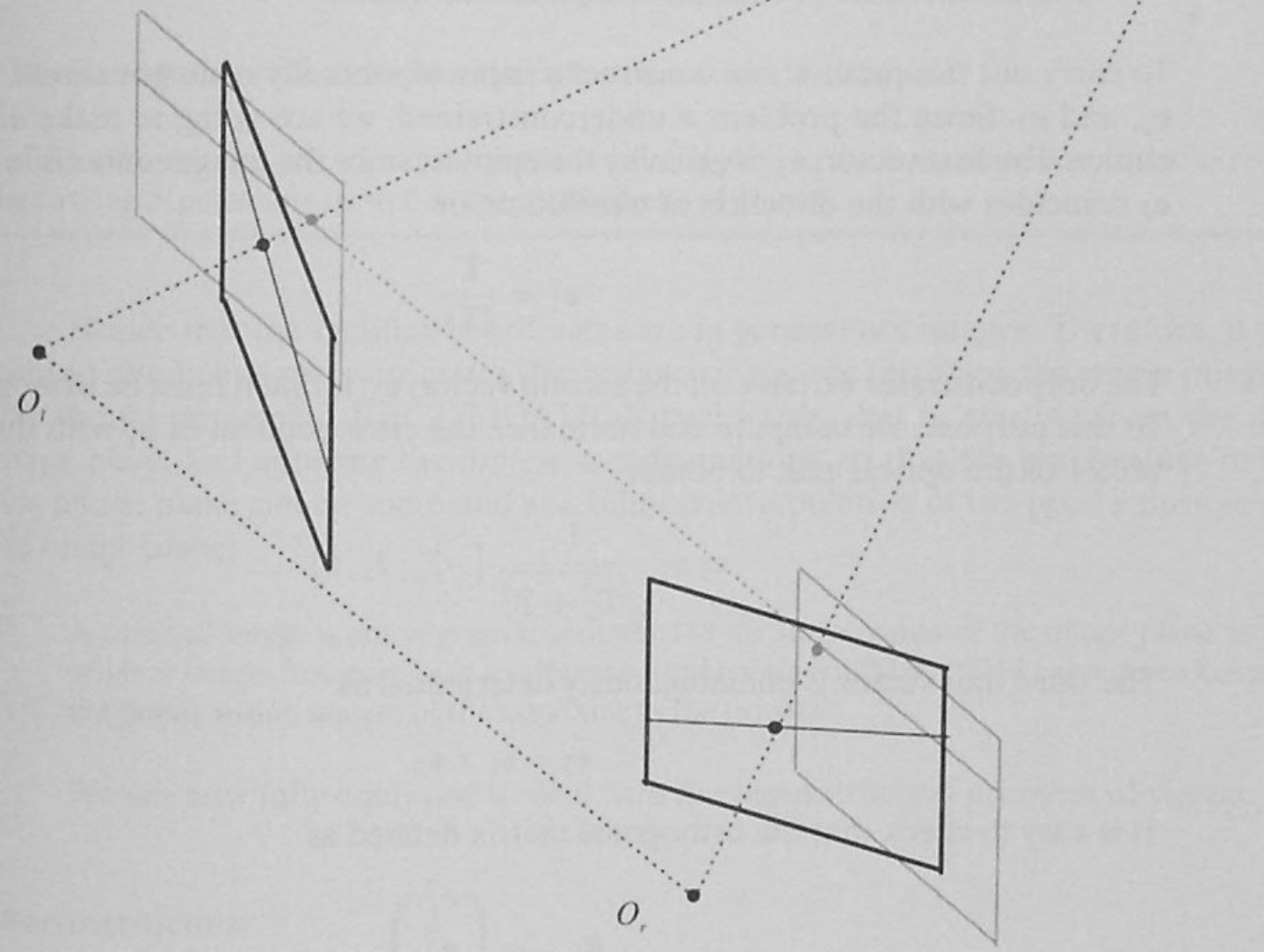
Alternatively, we can reconstruct a scene point using a purely algebraic approach: given the projection matrices  $\mathcal{M}$  and  $\mathcal{M}'$  and the matching points  $p$  and  $p'$ , we can rewrite the constraints  $z\mathbf{p} = \mathcal{M}\mathbf{P}$  and  $z'\mathbf{p}' = \mathcal{M}'\mathbf{P}$  as

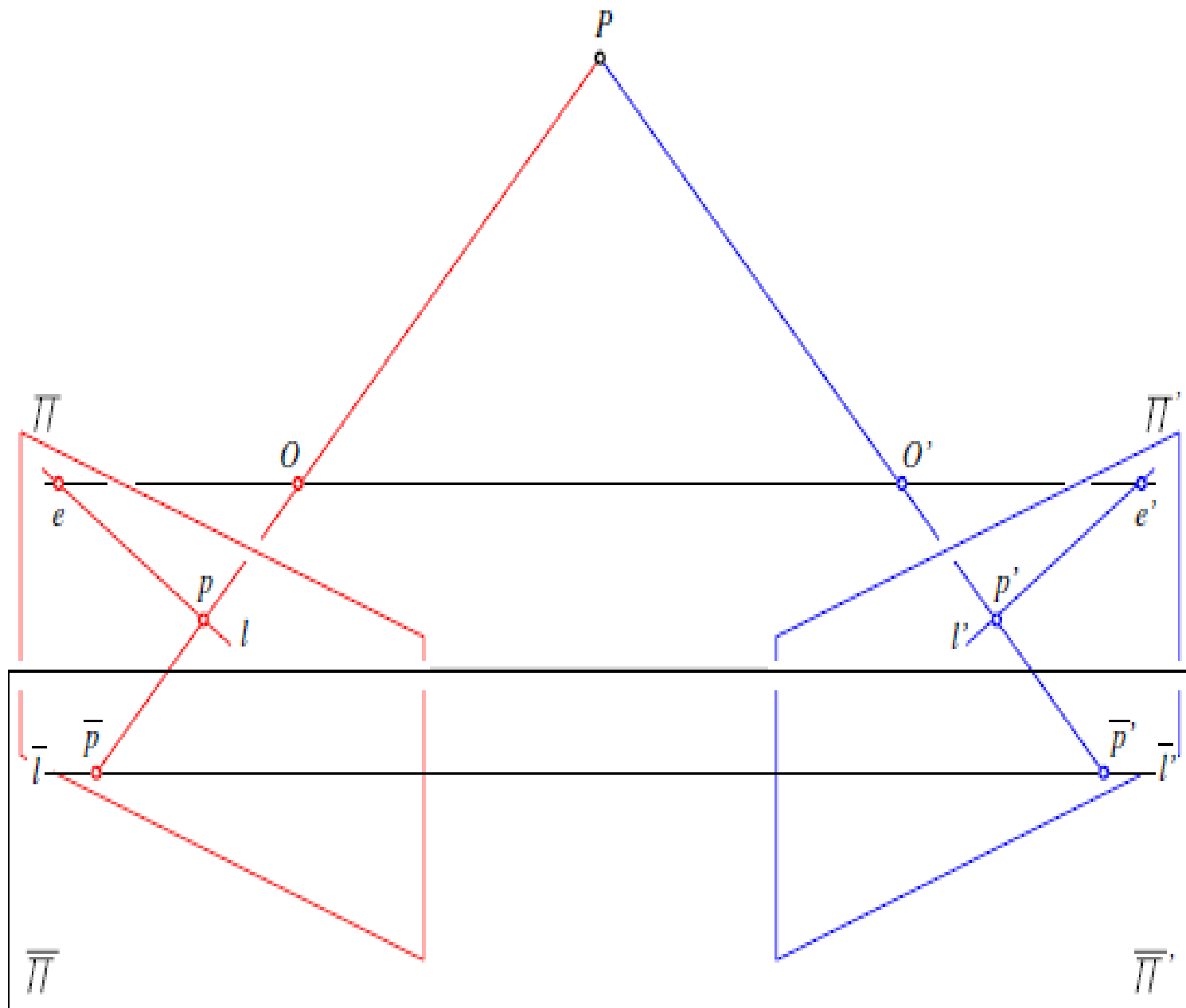
$$\begin{cases} \mathbf{p} \times \mathcal{M}\mathbf{P} = 0 \\ \mathbf{p}' \times \mathcal{M}'\mathbf{P} = 0 \end{cases} \iff \begin{pmatrix} [\mathbf{p}_\times]\mathcal{M} \\ [\mathbf{p}'_\times]\mathcal{M}' \end{pmatrix} \mathbf{P} = 0.$$

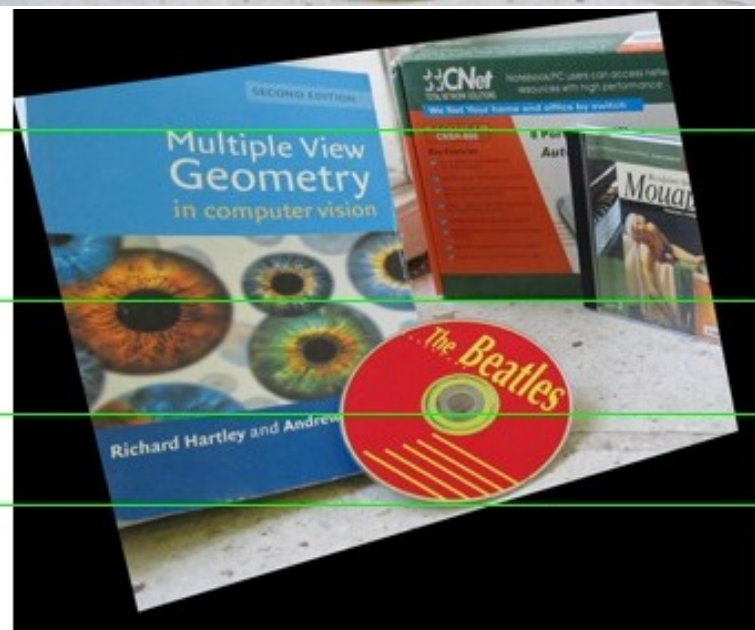
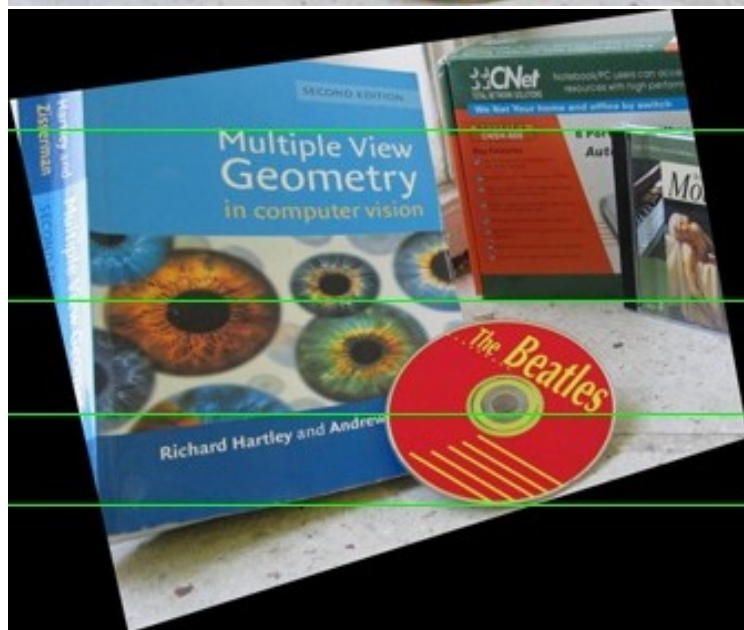
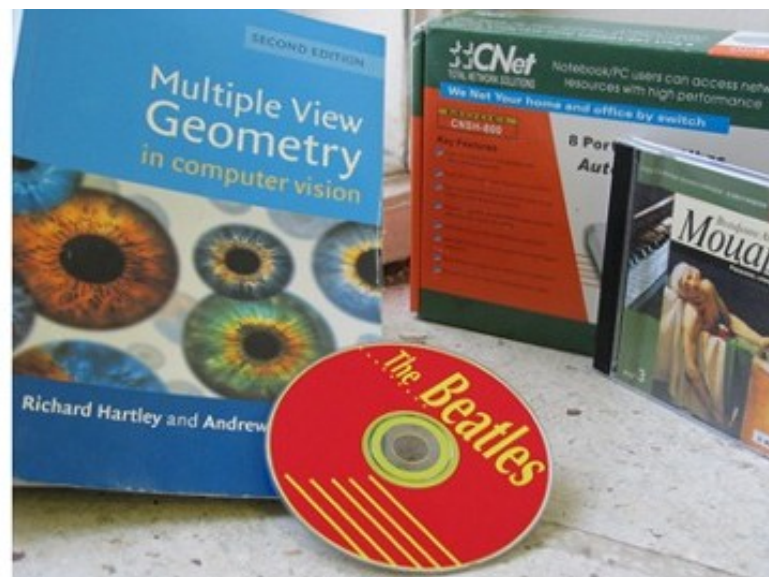
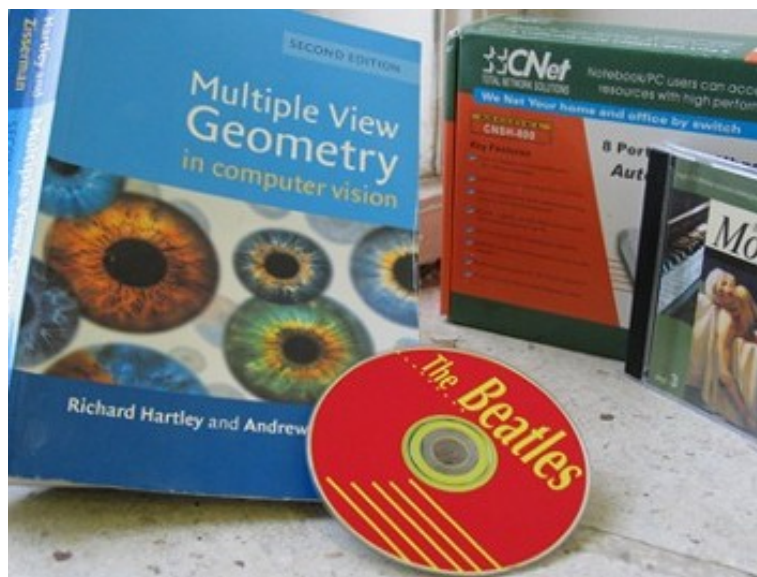
This is an overconstrained system of four independent linear equations in the homogeneous coordinates of  $P$ , that is easily solved using the linear least-squares

# Rectification

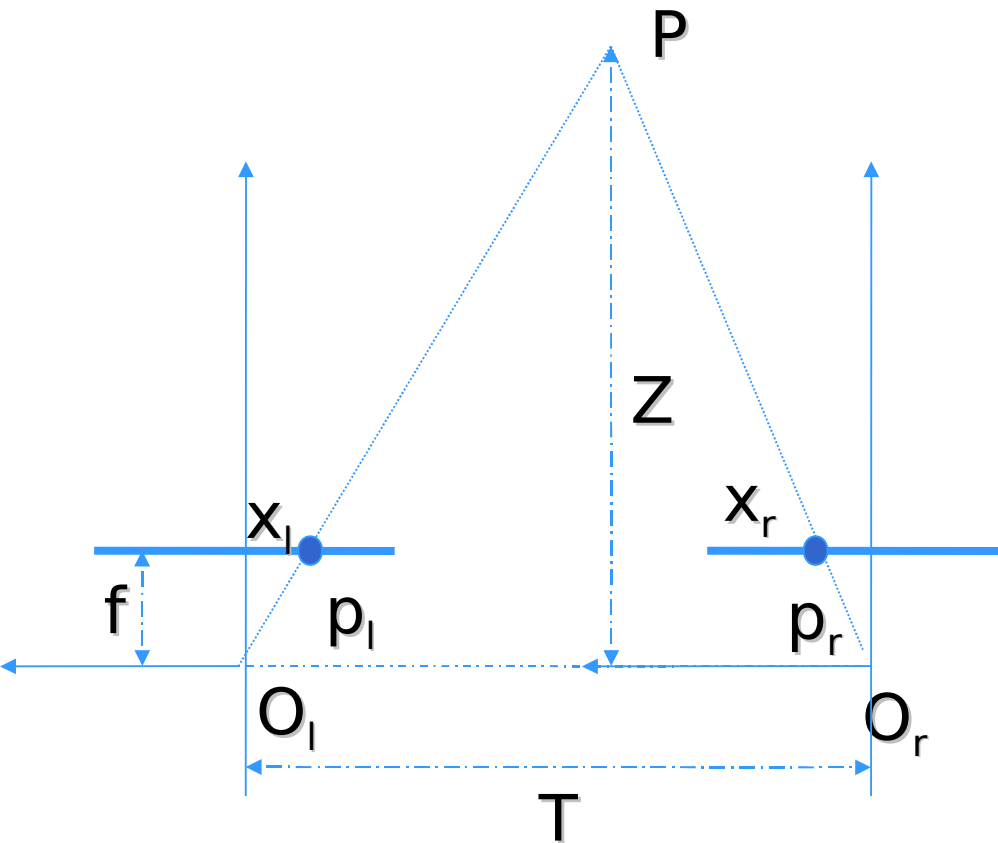
- The calculations associated with stereo algorithms are often considerably simplified when the images of interest have been rectified, i.e., replaced by two projectively
- equivalent pictures with a common image plane parallel to the baseline joining the two optical centers (Figure 13.5).
- The rectification process can be implemented by projecting the original pictures onto the new image plane.
- With an appropriate choice of coordinate system, the rectified epipolar lines are scanlines of the new images, and they are also parallel to the baseline.







# A simple stereo system



$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z}$$

$$Z = f \frac{T}{x_r - x_l}$$

Disparity:  $d = x_r - x_l$

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

**T** is the stereo baseline

**d** measures the difference in retinal position between corresponding points

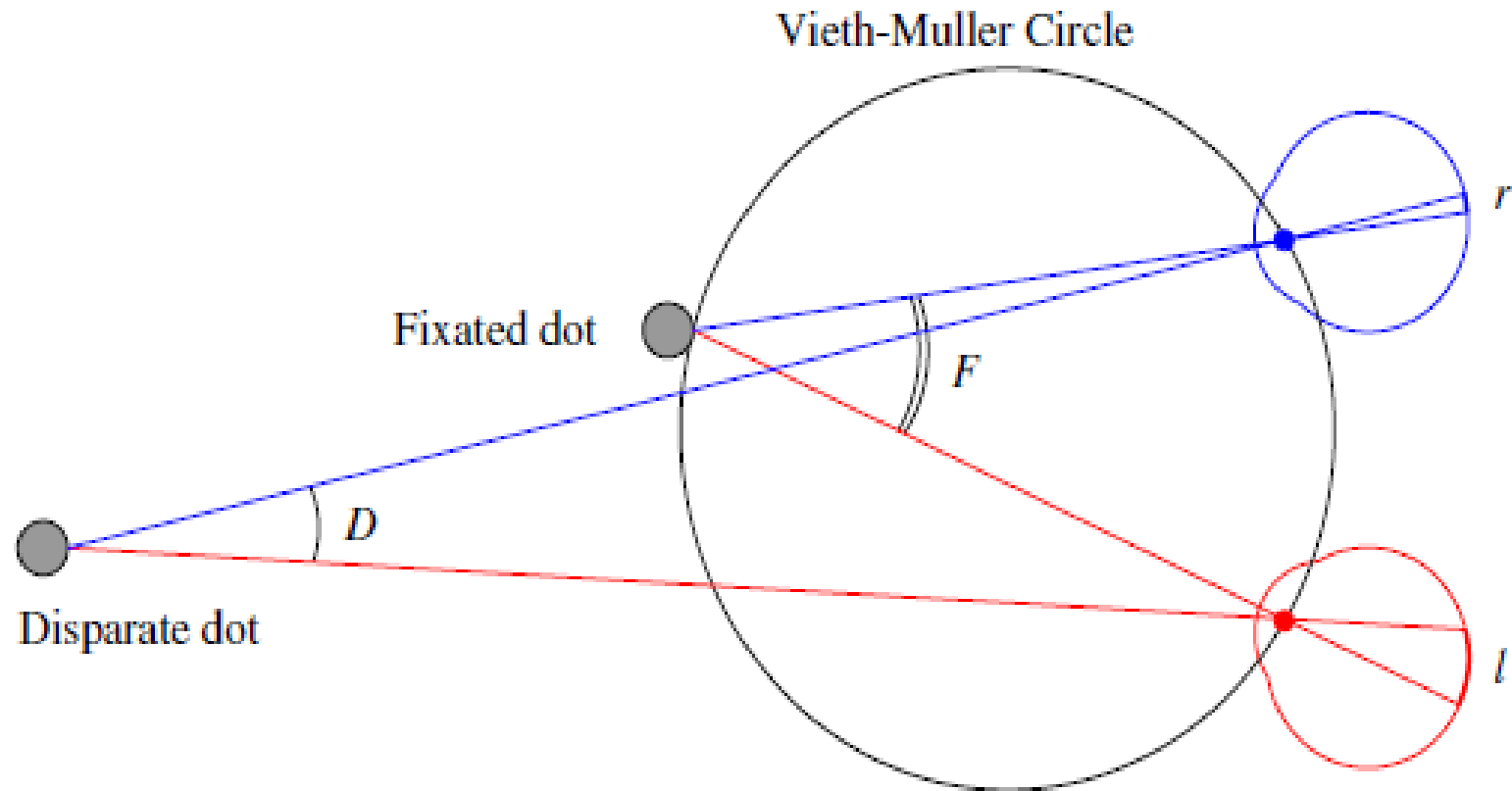
# 3D Reconstruction Problem

Three cases of 3D reconstruction depending on the amount of a priori knowledge on the stereo system

- **Both intrinsic and extrinsic known** - > can solve the reconstruction problem unambiguously by triangulation
- **Only intrinsic known** -> recovery structure and extrinsic up to an unknown scaling factor
- **Only correspondences** -> reconstruction only up to an unknown, global projective transformation



# Human Stereo



$$d = D - F,$$

# Stereo with Parallel Cameras

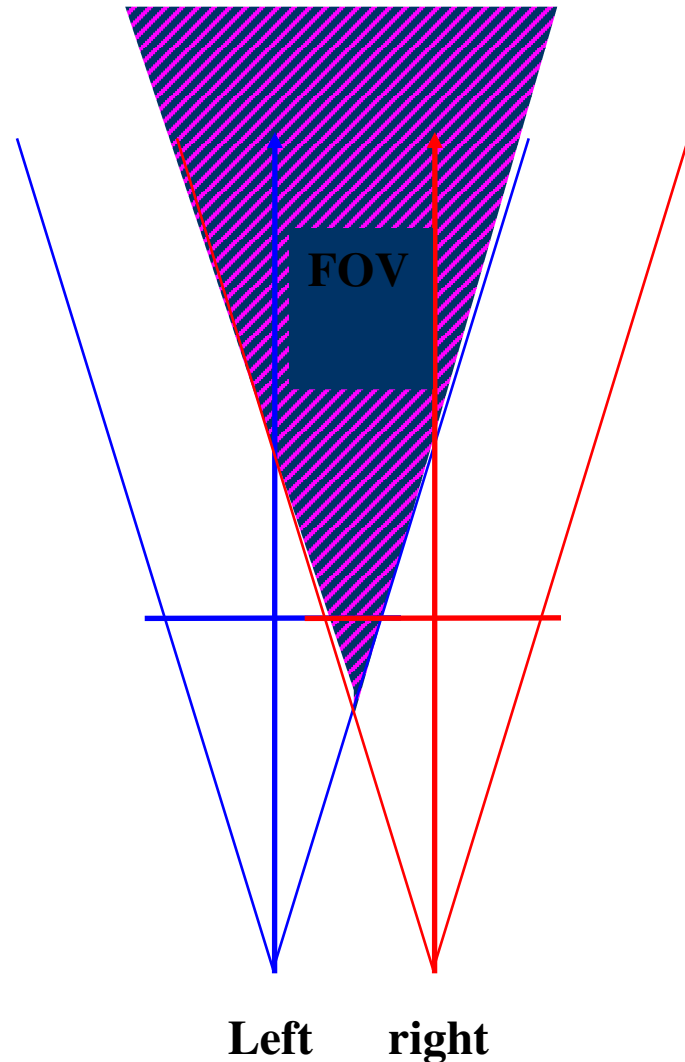
- Stereo with Parallel Axes

- Short baseline

- large common FOV
    - large depth error

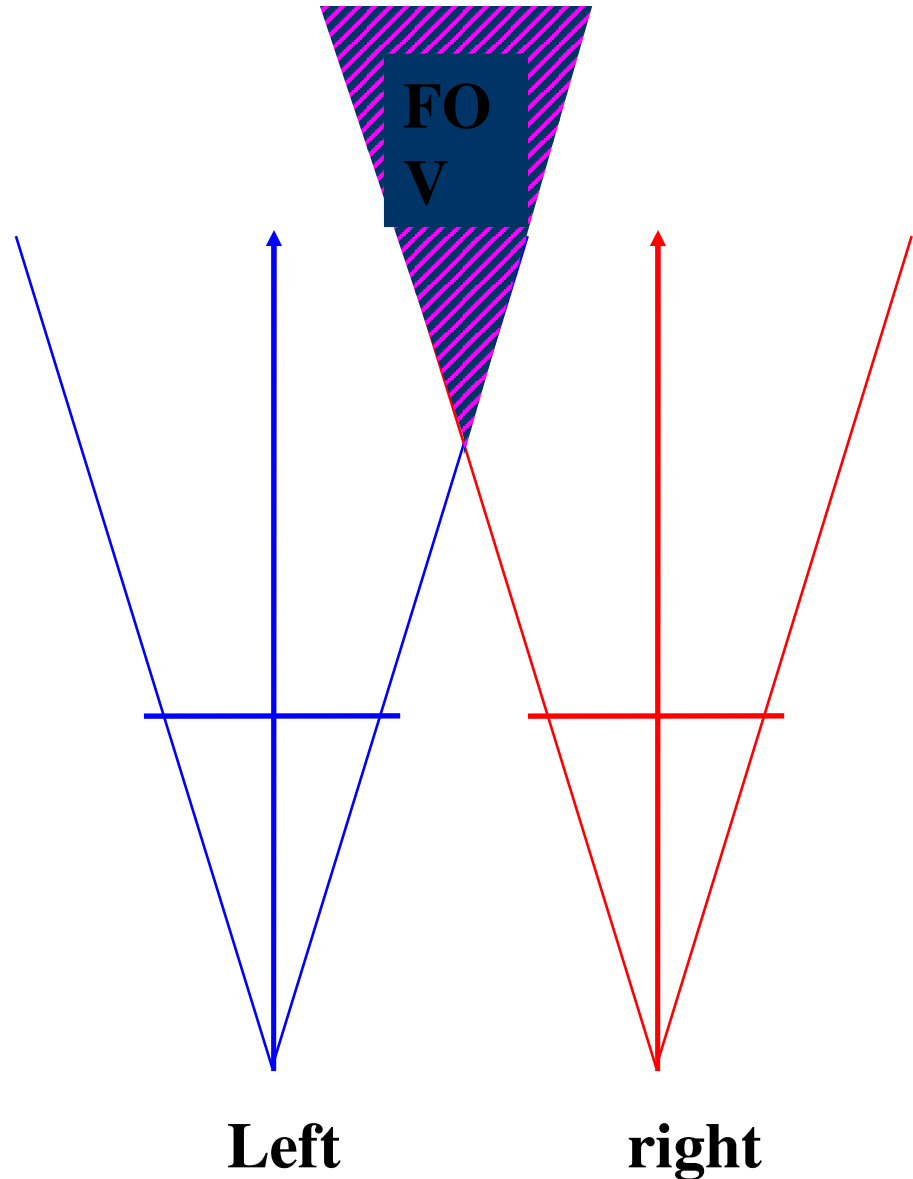
- Long baseline

- small depth error
    - small common FOV
    - More occlusion problems



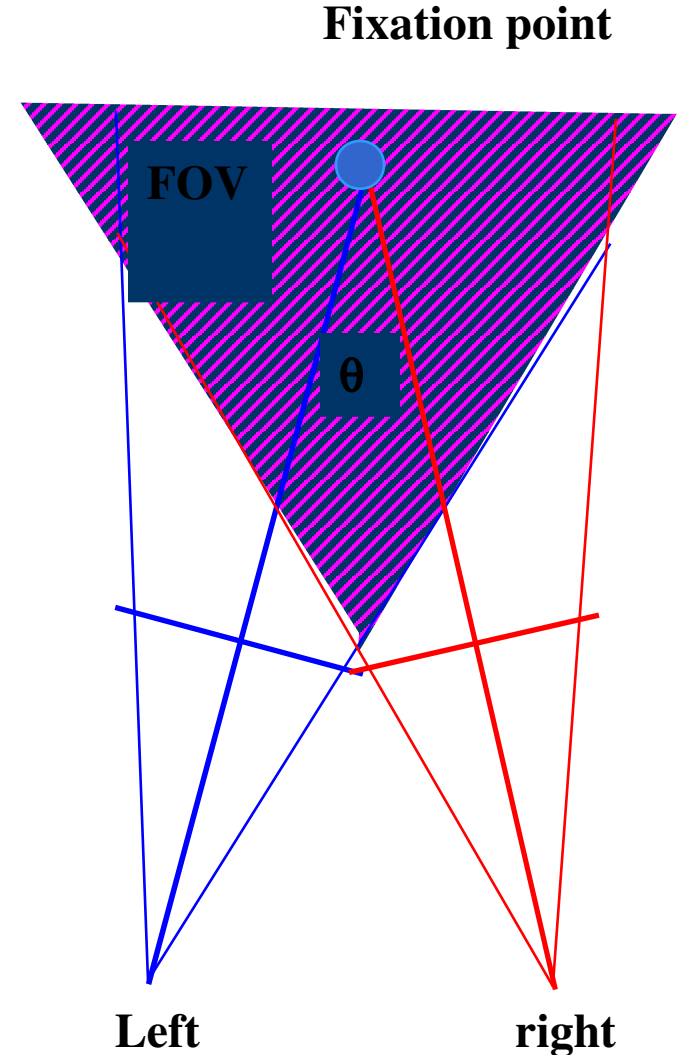
# Stereo with Parallel Cameras

- Stereo with Parallel Axes
  - Short baseline
    - large common FOV
    - large depth error
  - Long baseline
    - small depth error
    - small common FOV
    - More occlusion problems
- Depth Accuracy vs. Depth
  - Depth Error is proportional to  $\text{Depth}^2$
  - Nearer the point, better the depth estimation



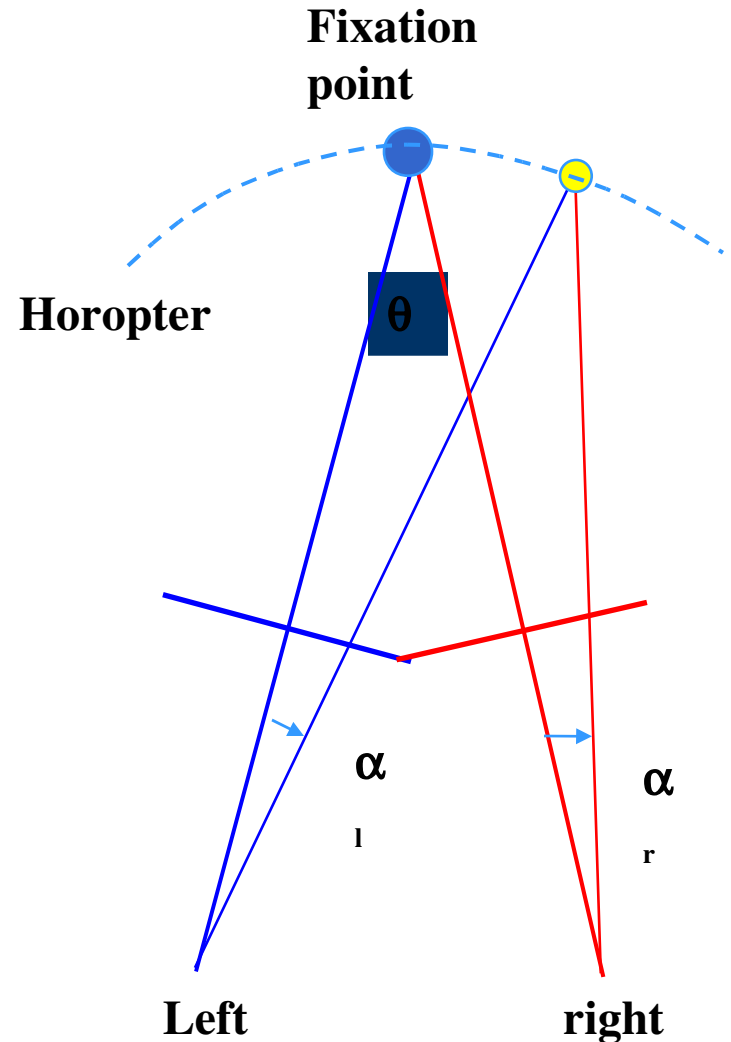
# Stereo with Converging Cameras

- Two optical axes intersect at the Fixation Point
  - converging angle  $\theta$
  - The common FOV Increases



# Stereo with Converging Cameras

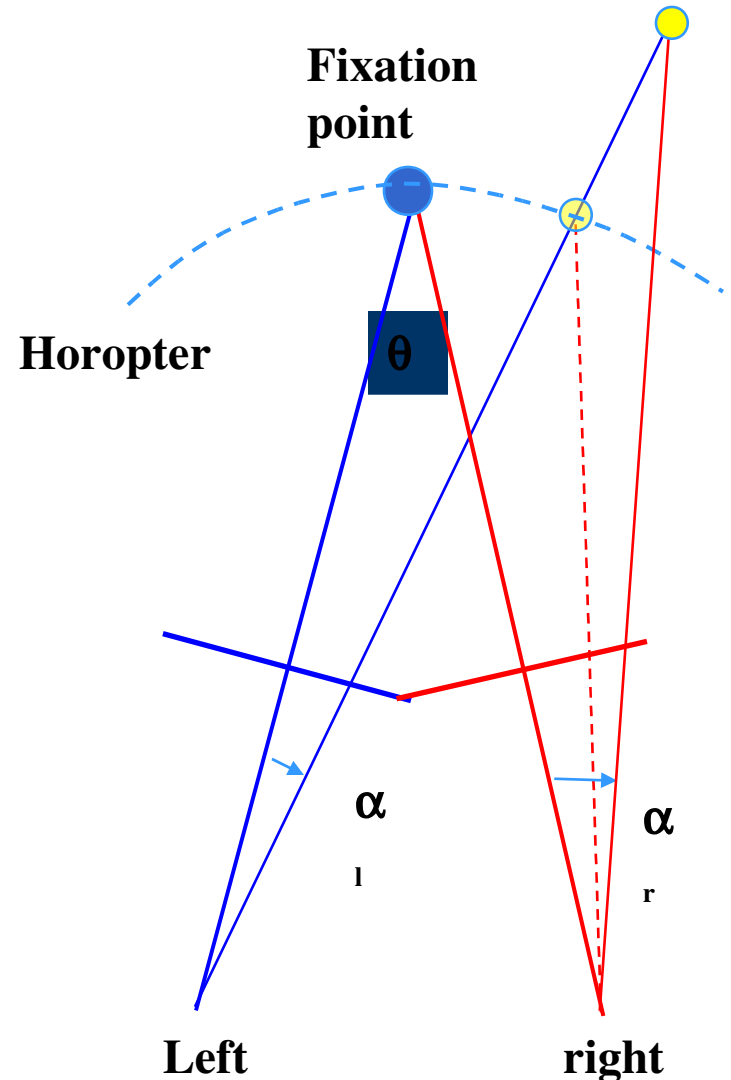
- Disparity properties
  - Disparity uses angle instead of distance
  - Zero disparity at fixation point
    - and the Zero-disparity horopter



# Stereo with Converging Cameras

## Disparity properties

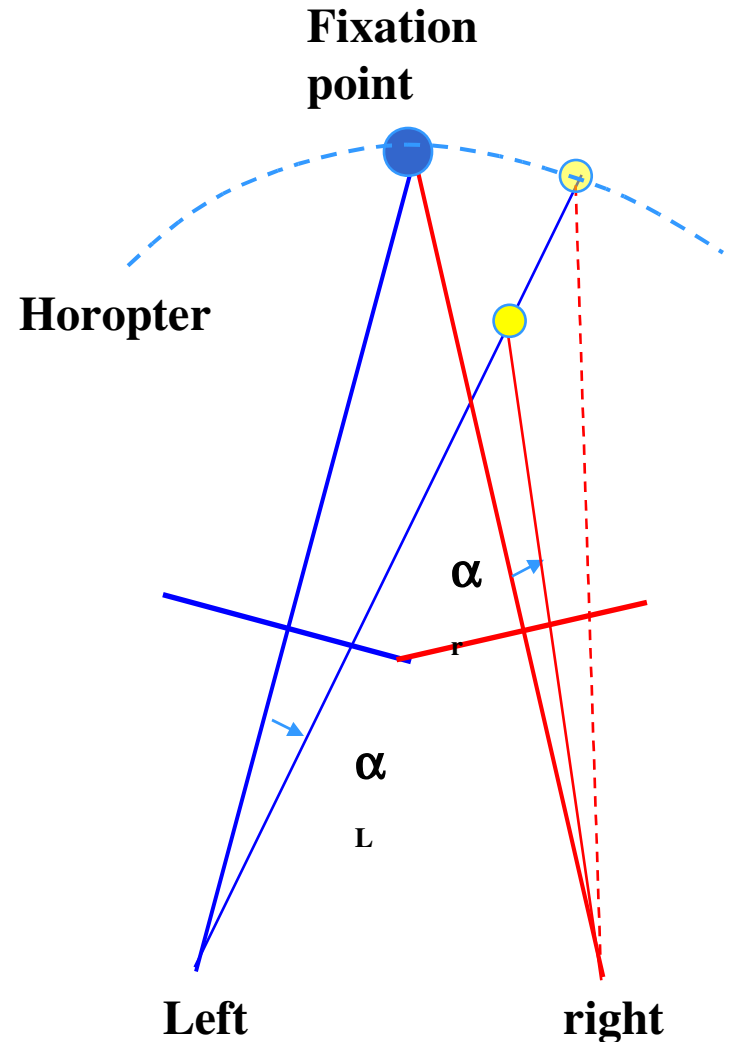
- Disparity uses angle instead of distance
- Zero disparity at fixation point
  - and the Zero-disparity horopter
- Disparity increases with the distance of objects from the fixation points
  - $>0$  : outside of the horopter
  - $<0$  : inside the horopter



# Stereo with Converging Cameras

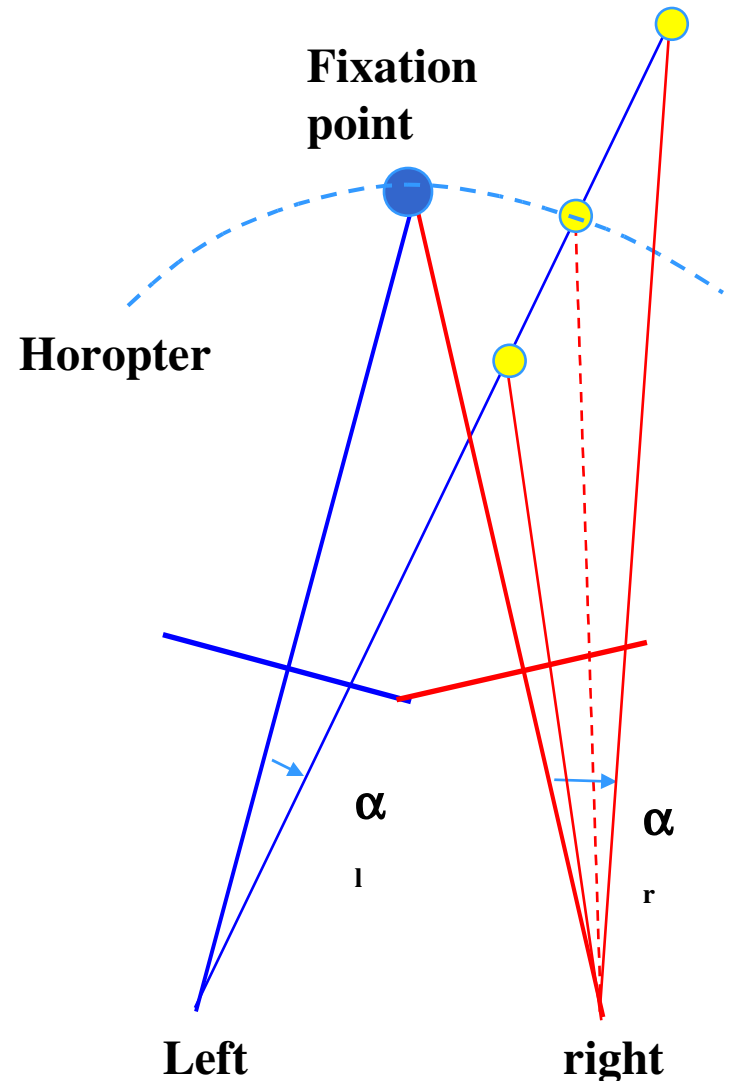
## Disparity properties

- Disparity uses angle instead of distance
- Zero disparity at fixation point
  - and the Zero-disparity horopter
- Disparity increases with the distance of objects from the fixation points
  - $>0$  : outside of the horopter
  - $<0$  : inside the horopter



# Stereo with Converging Cameras

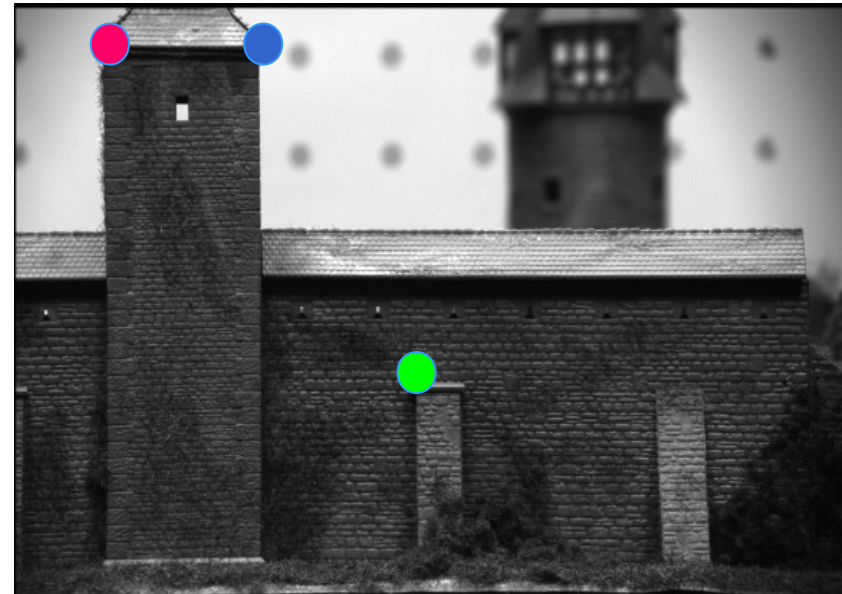
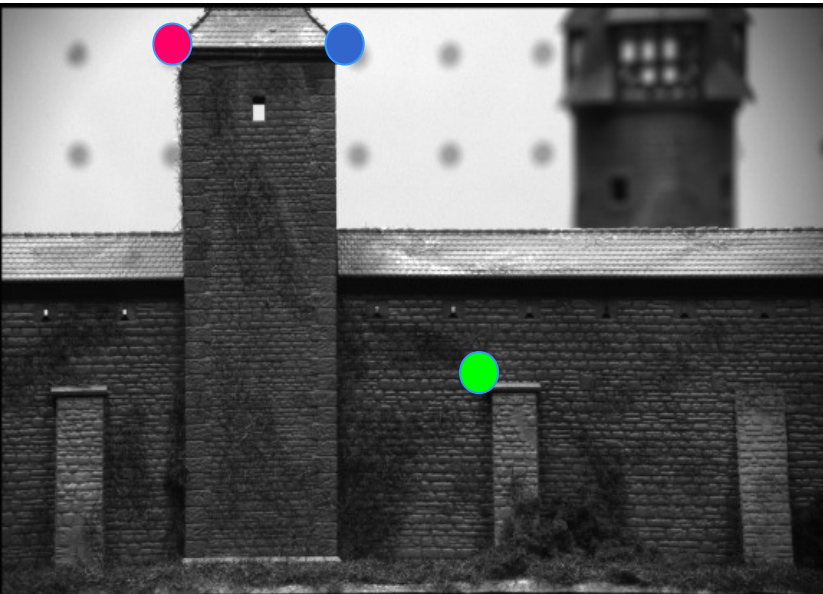
- Disparity properties
  - Disparity uses angle instead of distance
  - Zero disparity at fixation point
    - and the Zero-disparity horopter
  - Disparity increases with the distance of objects from the fixation points
    - $>0$  : outside of the horopter
    - $<0$  : inside the horopter
- Depth Accuracy vs. Depth
  - Depth Error is proportional to  $\text{Depth}^2$
  - Nearer the point, better the depth estimation



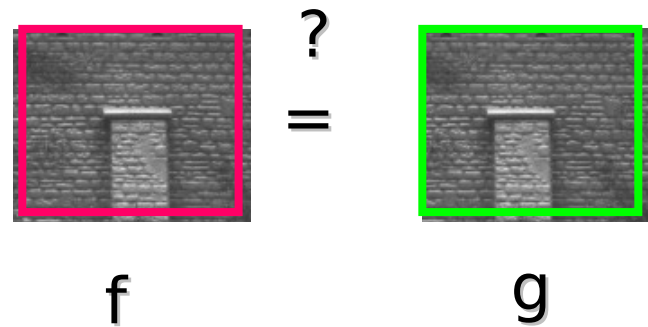


# Finding Correspondences

- We need to compare windows



# Comparing Windows:



Some possible measures:

$$\max_{[i,j] \in R} |f(i, j) - g(i, j)|$$

$$\sum_{[i,j] \in R} |f(i, j) - g(i, j)|$$

$$SSD = \sum_{[i,j] \in R} (f(i, j) - g(i, j))^2$$

$$C_{fg} = \sum_{[i,j] \in R} f(i, j)g(i, j)$$

Most  
popular

“SSD” or “block matching”  
(Sum of Squared Differences)

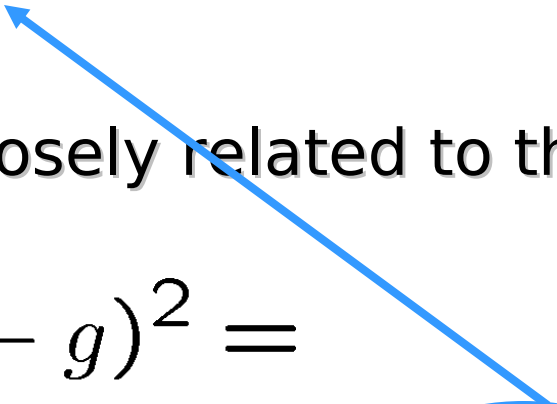
$$\sum_{[i,j] \in R} (f(i, j) - g(i, j))^2$$

It is the most popular.

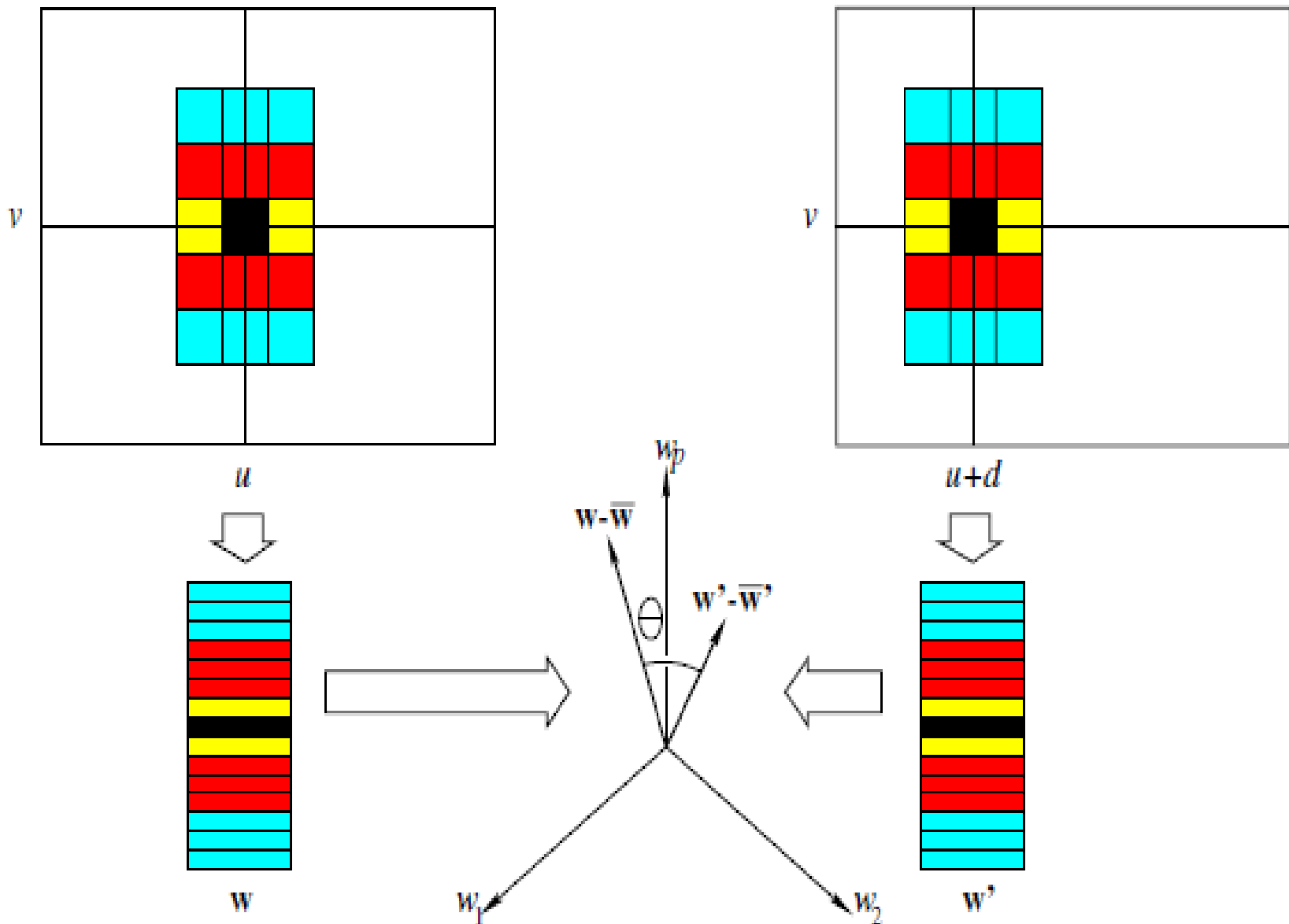
# Cross-Correlation $C_{fg}$

$$C_{fg} = \sum_{[i,j] \in R} f(i,j)g(i,j)$$

Is also very popular and it is closely related to the SSD:

$$\begin{aligned} SSD &= \sum_{[i,j] \in R} (f - g)^2 = \\ &= \sum_{[i,j] \in R} f^2 + \sum_{[i,j] \in R} g^2 - 2 \sum_{[i,j] \in R} fg \end{aligned}$$


# Normalized Cross Correlation



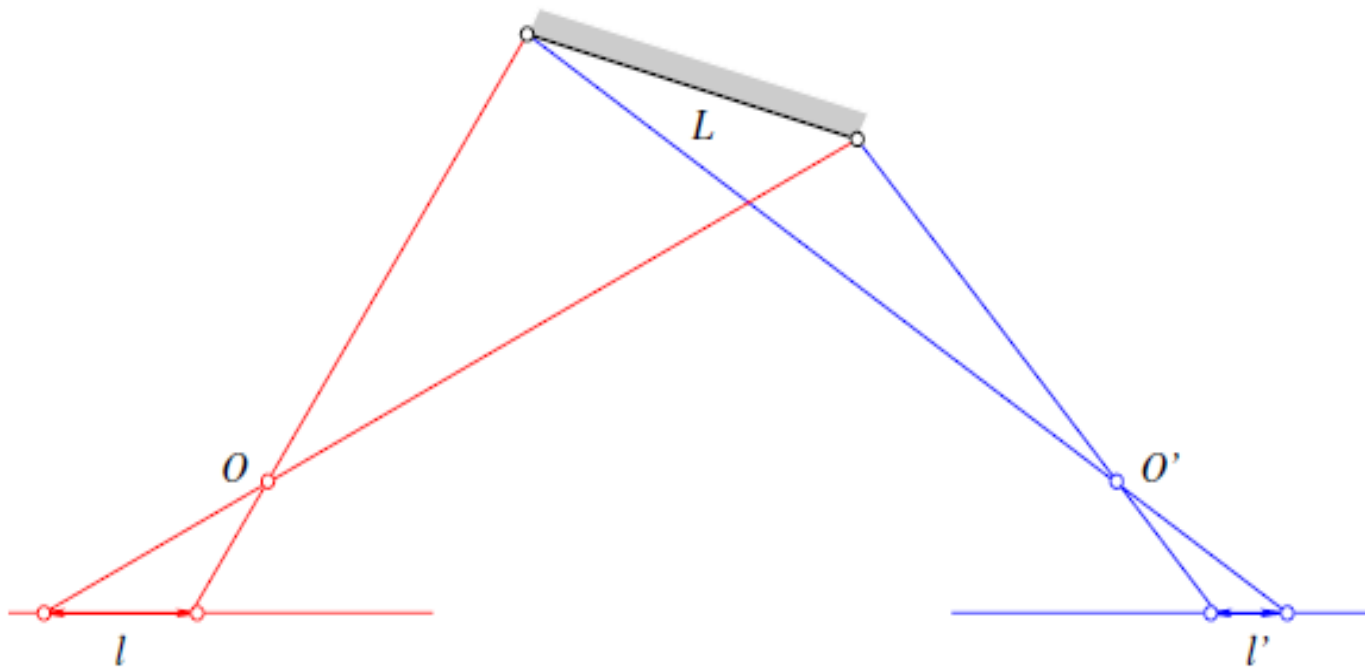
$$NCC = \sum_{i=1}^n \frac{(x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y}$$

# Feature-based Methods

- Conceptually very similar to Correlation-based methods, but:
  - They only search for correspondences of a sparse set of image features.
  - Correspondences are given by the most similar feature pairs.
  - Similarity measure must be adapted to the type of feature used.

# MultiScale Mathcing

- Correlation has problems in slanted surfaces

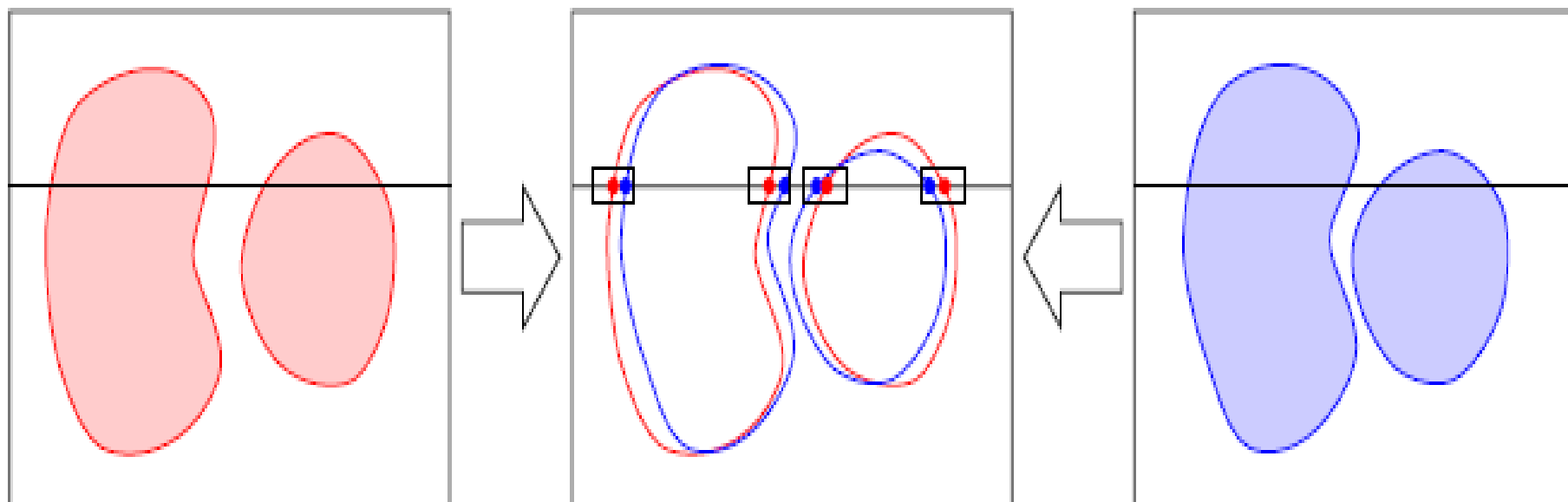


**Figure 13.12.** The foreshortening of non-frontoparallel surfaces is different for the two cameras: a surface segment with length  $L$  projects onto two image segments of different lengths  $l$  and  $l'$ .

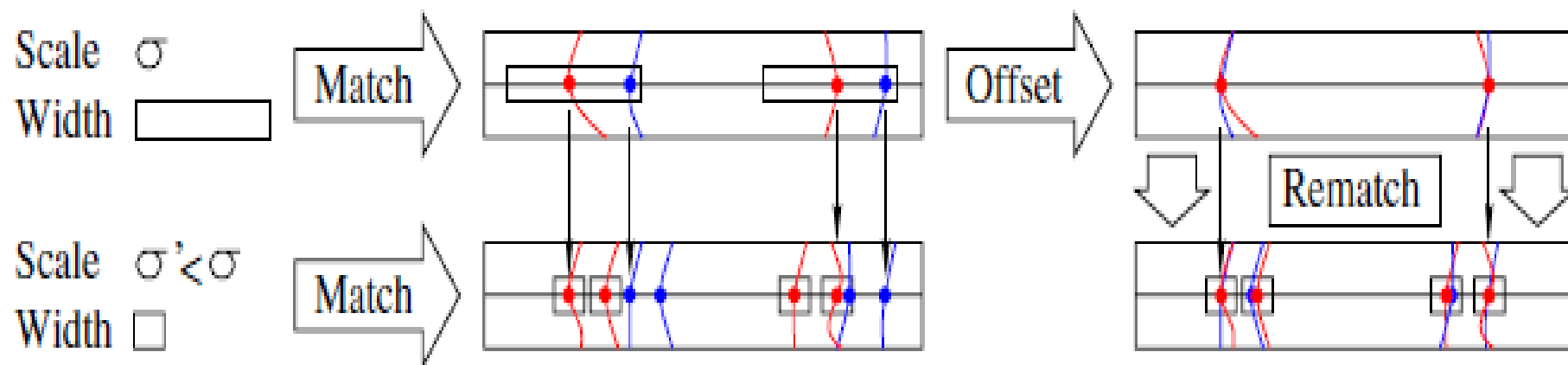


# MultiScale Edge Matching

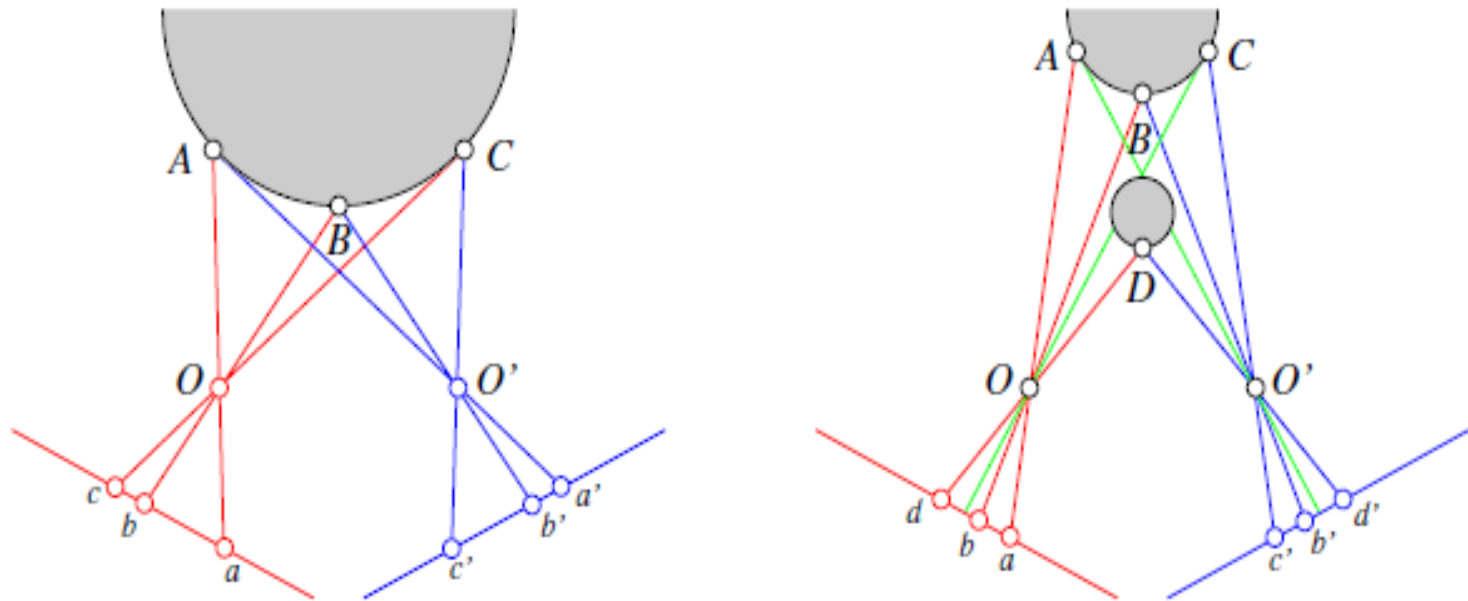
1. Convolve the two (rectified) images with  $\nabla^2 G_\sigma$  filters of increasing standard deviations  $\sigma_1 < \sigma_2 < \sigma_3 < \sigma_4$ .
2. Find zero crossings of the Laplacian along horizontal scanlines of the filtered images.
3. For each filter scale  $\sigma$ , match zero crossings with the same parity and roughly equal orientations in a  $[-w_\sigma, +w_\sigma]$  disparity range, with  $w_\sigma = 2\sqrt{2}\sigma$ .
4. Use the disparities found at larger scales to control eye vergence and cause unmatched regions at smaller scales to come into correspondence.



## Matching zero-crossings at multiple scales

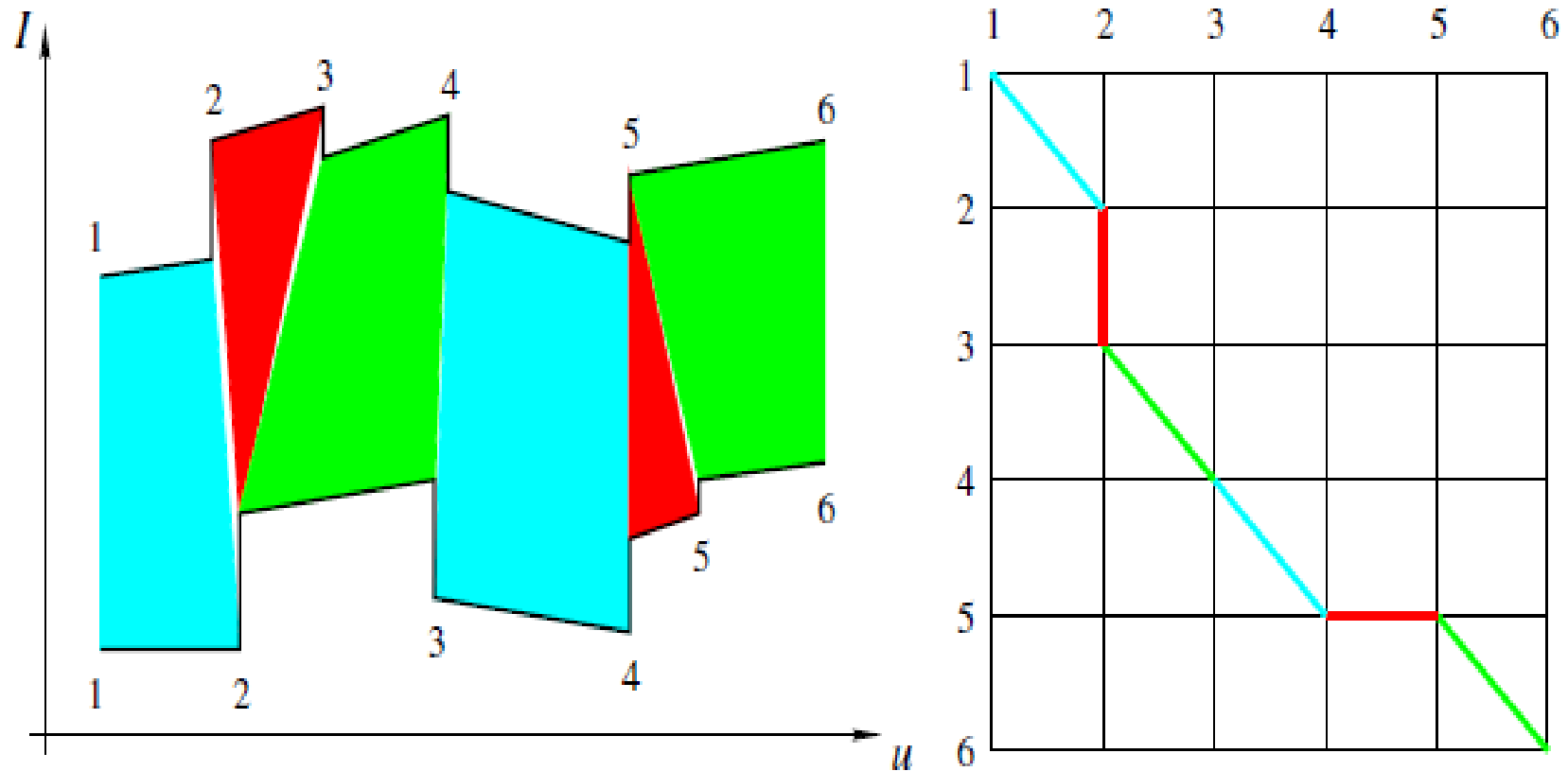


# Ordering Constraint



**Figure 13.16.** Ordering constraints. In the (usual) case shown in the left part of the diagram, the order of feature points along the two (oriented) epipolar lines is the same, and it is the inverse of the order of the scene points along the curve where the observed surface intersects the epipolar plane. In the case shown in the right part of the figure, a small object lies in front of a larger one. Some of the surface points are not visible in one of the images (e.g.,  $A$  is not visible in the right image), and the order of the image points is not the same in the two pictures:  $b$  is on the right of  $d$  in the left image, but  $b'$  is on the left of  $d'$  in the right image.

# Ordering Constraint



**Figure 13.17.** Dynamic programming and stereopsis: the left part of the figure shows two intensity profiles along matching epipolar lines. The polygons joining the two profiles indicate matches between successive intervals (some of the matched intervals may have zero length). The right part of the diagram represents the same information in graphical form: an arc (thick line segment) joins two nodes  $(i, i')$  and  $(j, j')$  when the intervals  $(i, j)$  and  $(i', j')$  of the intensity profiles match each other.

# Which method should we use?

- Correlation methods:
  - dense maps, good for surface reconstruction
  - Require textured images
  - Sensitive to illumination variations
  - Inadequate for very different viewpoints
- Feature methods:
  - Sparse maps, good for navigation
  - Require prior knowledge of type of scene
  - Must find features first