Unit 4
Stereo Vision

Ref: Szeliski, Sec. 6.2, 6.3, 7.1, 7.2

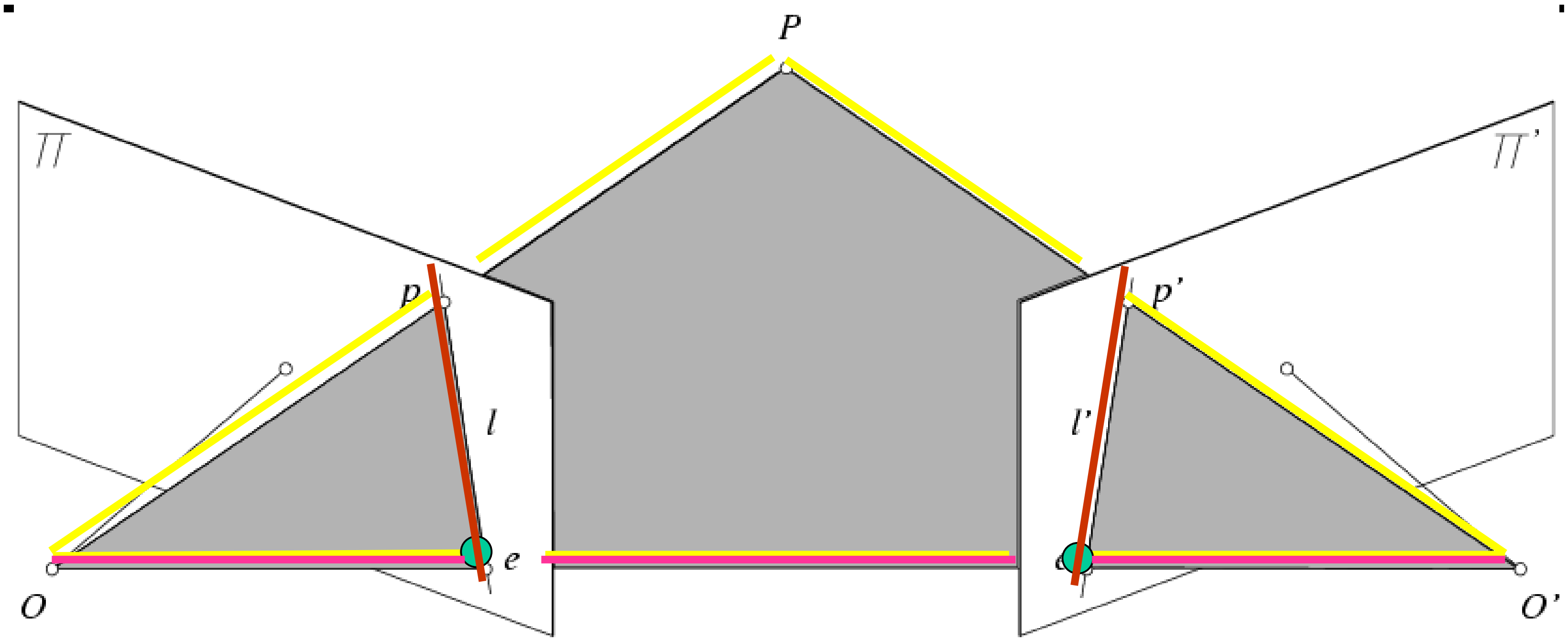
Review

- Camera Model
- Camera Calibration
- Image Warping
- Stereo Geometry

Today

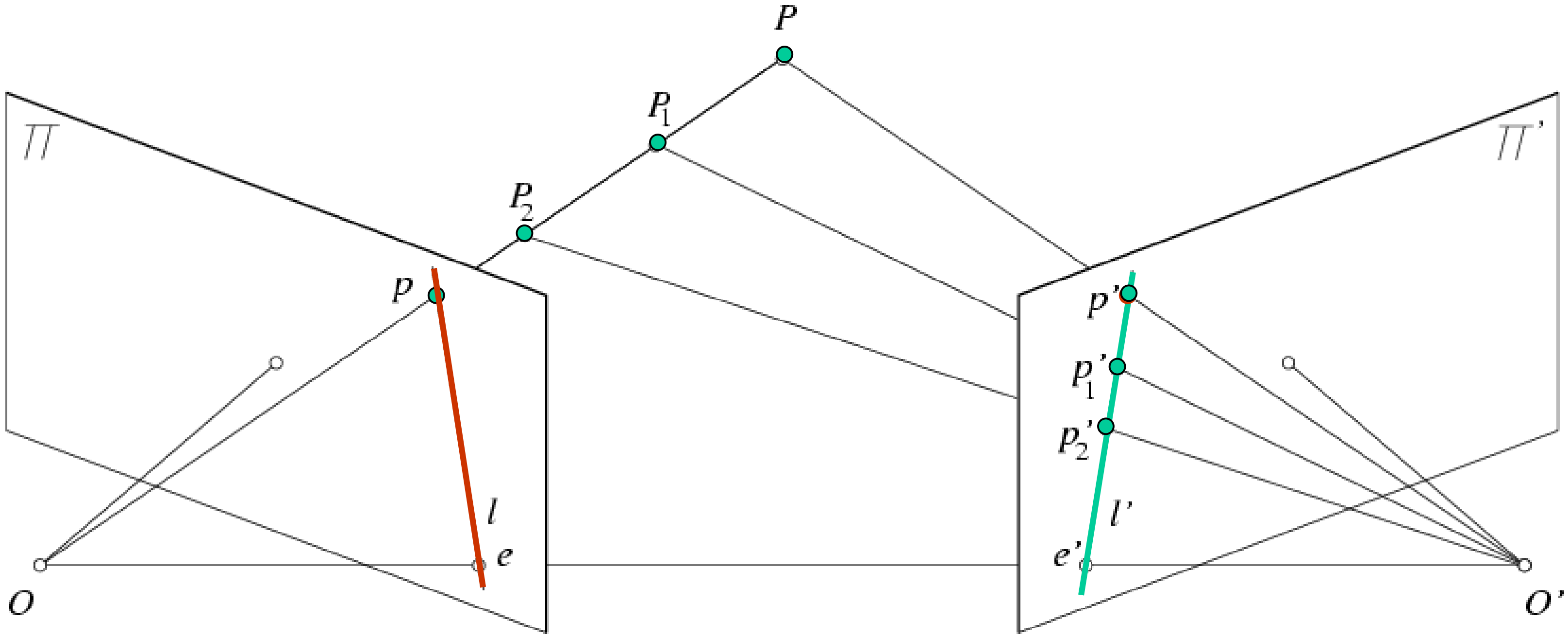
- Epipolar geometry
- Stereo Matching

Epipolar Geometry



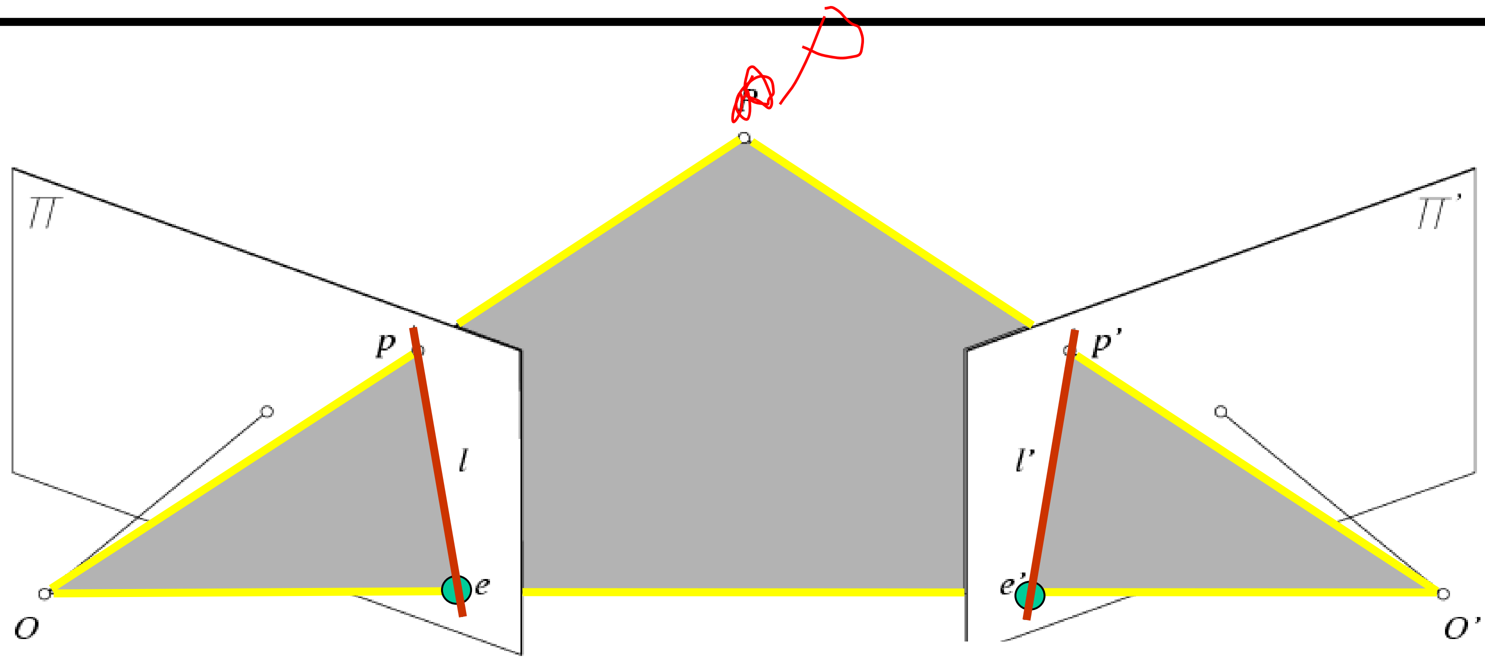
- Epipolar Plane
- Baseline
- Epipoles
- Epipolar Lines

Epipolar Constraint



- Potential matches for *p* have to lie on the corresponding epipolar line *l′*.
- Potential matches for *p′* have to lie on the corresponding epipolar line *l*.

Epipolar Constraint: Calibrated Case



$$\overrightarrow{Op} \cdot [\overrightarrow{OO'} \times \overrightarrow{O'p'}] = 0 \quad \Rightarrow \quad \boldsymbol{p} \cdot [\boldsymbol{t} \times (\mathcal{R}\boldsymbol{p}')] = 0 \quad \text{with} \begin{cases} \boldsymbol{p} = (u, v, 1)^T \\ \boldsymbol{p}' = (u', v', 1)^T \\ \mathcal{M} = (\text{Id} \quad \boldsymbol{0}) \\ \mathcal{M}' = (\mathcal{R}^T, -\mathcal{R}^T\boldsymbol{t}) \end{cases}$$

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

Essential Matrix
(Longuet-Higgins, 1981)

Cross Product

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

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

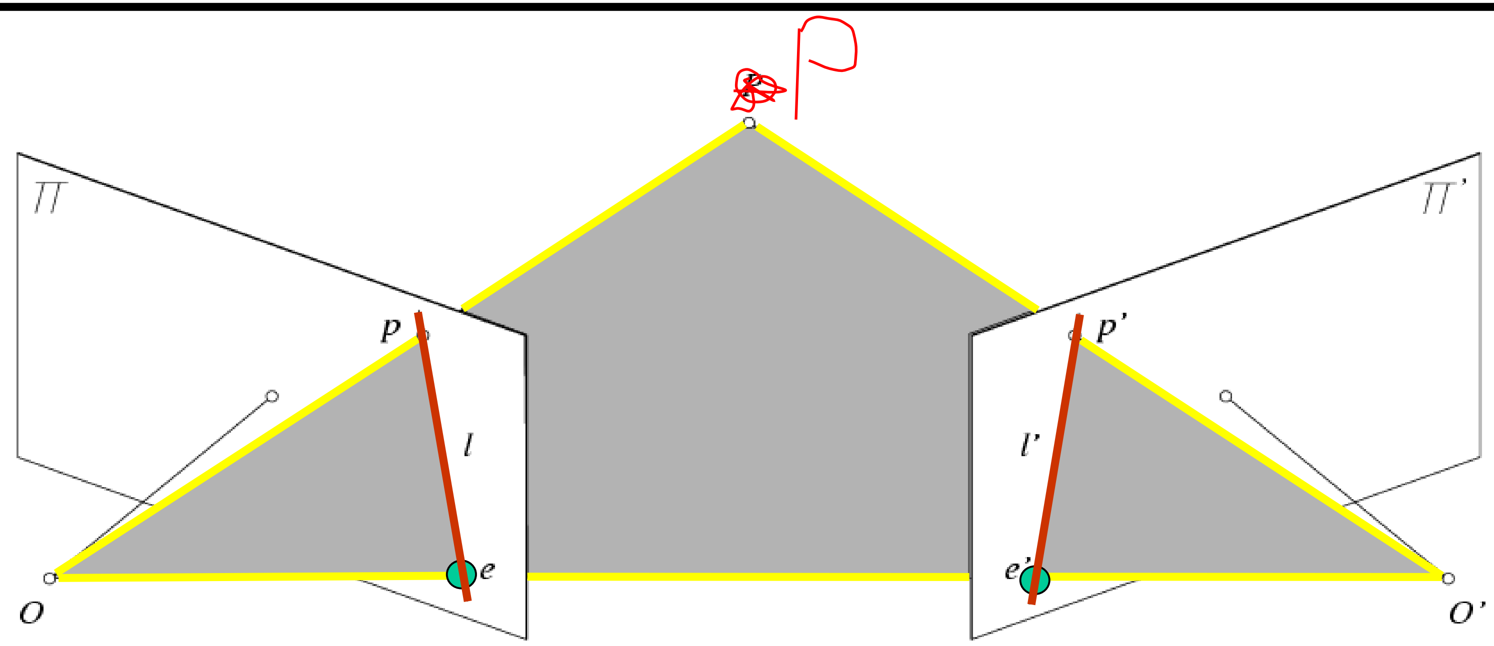
$$[\mathbf{a}]_{\times} \stackrel{\text{def}}{=} \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

Properties of the Essential Matrix

$$\boxed{\boldsymbol{p}^T \mathcal{E} \boldsymbol{p}' = 0 \quad \text{with} \quad \mathcal{E} = [\boldsymbol{t}_\times]\mathcal{R}}$$

- $\mathcal{E}$ p′ is the epipolar line associated with p′.
- $\mathcal{E}^T$p is the epipolar line associated with p.
- $\mathcal{E}$ e′=0 and $\mathcal{E}^T$e=0.
- $\mathcal{E}$ is singular.
- $\mathcal{E}$ has two equal non-zero singular values (Huang and Faugeras, 1989).

Epipolar Constraint: Uncalibrated Case



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

$$\mathbf{p} = \mathcal{K}\hat{\mathbf{p}}$$

$$\mathbf{p}' = \mathcal{K}'\hat{\mathbf{p}}'$$

$$\mathbf{p}^T \mathcal{F} \mathbf{p}' = 0 \quad \text{with} \quad \mathcal{F} = \mathcal{K}^{-T} \mathcal{E} \mathcal{K}'^{-1}$$

Fundamental Matrix
(Faugeras and Luong, 1992)

Properties of the Fundamental Matrix

- $\mathcal{F}\, p'$ is the epipolar line associated with p'.
- $\mathcal{F}^T p$ is the epipolar line associated with p.
- $\mathcal{F}\, e'=0$ and $\mathcal{F}^T e=0$.
- $\mathcal{F}$ is singular.

The Eight-Point Algorithm (Longuet-Higgins, 1981)

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

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

$$\begin{pmatrix} u_1u_1' & u_1v_1' & u_1 & v_1u_1' & v_1v_1' & v_1 & u_1' & v_1' \\ u_2u_2' & u_2v_2' & u_2 & v_2u_2' & v_2v_2' & v_2 & u_2' & v_2' \\ u_3u_3' & u_3v_3' & u_3 & v_3u_3' & v_3v_3' & v_3 & u_3' & v_3' \\ u_4u_4' & u_4v_4' & u_4 & v_4u_4' & v_4v_4' & v_4 & u_4' & v_4' \\ u_5u_5' & u_5v_5' & u_5 & v_5u_5' & v_5v_5' & v_5 & u_5' & v_5' \\ u_6u_6' & u_6v_6' & u_6 & v_6u_6' & v_6v_6' & v_6 & u_6' & v_6' \\ u_7u_7' & u_7v_7' & u_7 & v_7u_7' & v_7v_7' & v_7 & u_7' & v_7' \\ u_8u_8' & u_8v_8' & u_8 & v_8u_8' & v_8v_8' & v_8 & u_8' & v_8' \end{pmatrix}\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{pmatrix} = -\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$\sum_{i=1}^{n} (\boldsymbol{p}_i^T \mathcal{F} \boldsymbol{p}_i')^2$$

under the constraint

$$|\mathcal{F}|^2 = 1.$$

Non-Linear Least-Squares Approach (Luong et al., 1993)

Minimize

$$\sum_{i=1}^{n}\left[\mathrm{d}^2(\boldsymbol{p}_i, \mathcal{F}\boldsymbol{p}'_i) + \mathrm{d}^2(\boldsymbol{p}'_i, \mathcal{F}^T\boldsymbol{p}_i)\right]$$

with respect to the coefficients of $\mathcal{F}$, using an appropriate rank-2 parameterization.

Problem with eight-point algorithm

250906.36	183269.57	921.81	200931.10	146766.13	738.21	272.19	198.81	1.00
2692.28	131633.03	176.27	6196.73	302975.59	405.71	15.27	746.79	1.00
416374.23	871684.30	935.47	408110.89	854384.92	916.90	445.10	931.81	1.00
191183.60	171759.40	410.27	416435.62	374125.90	893.65	465.99	418.65	1.00
48988.86	30401.76	57.89	298604.57	185309.58	352.87	846.22	525.15	1.00
164786.04	546559.67	813.17	1998.37	6628.15	9.86	202.65	672.14	1.00
116407.01	2727.75	138.89	169941.27	3982.21	202.77	838.12	19.64	1.00
135384.58	75411.13	198.72	411350.03	229127.78	603.79	681.28	379.48	1.00

$$\begin{pmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{pmatrix} = 0$$

linear least-squares:
unit norm vector F yielding smallest residual

What happens when there is noise?

The Normalized Eight-Point Algorithm (Hartley, 1995)

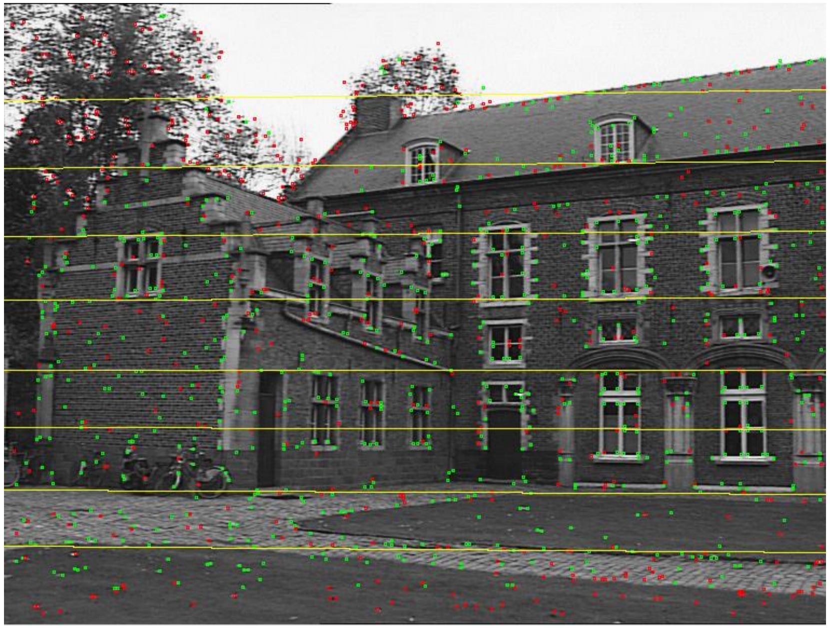
- Center the image data at the origin, and scale it so the mean squared distance between the origin and the data points is 2 pixels:
  $q_i = T\, p_i \quad , \quad q_i' = T'\, p_i'.$

- Use the eight-point algorithm to compute $\mathcal{F}$ from the points q_i and q_i' .

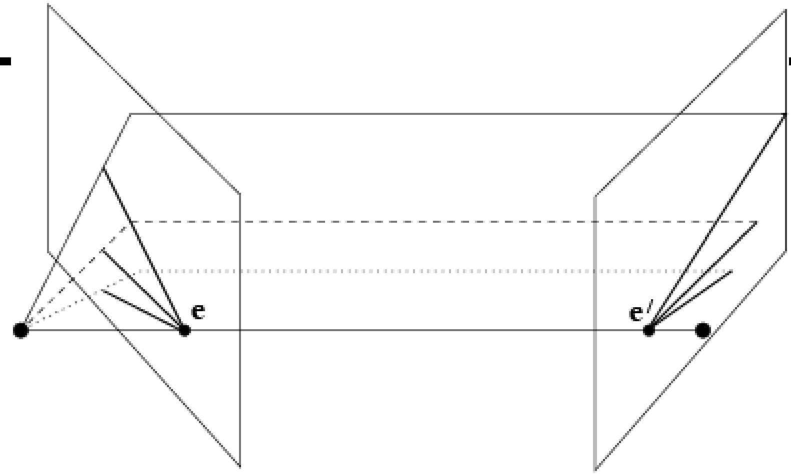
- Enforce the rank-2 constraint.

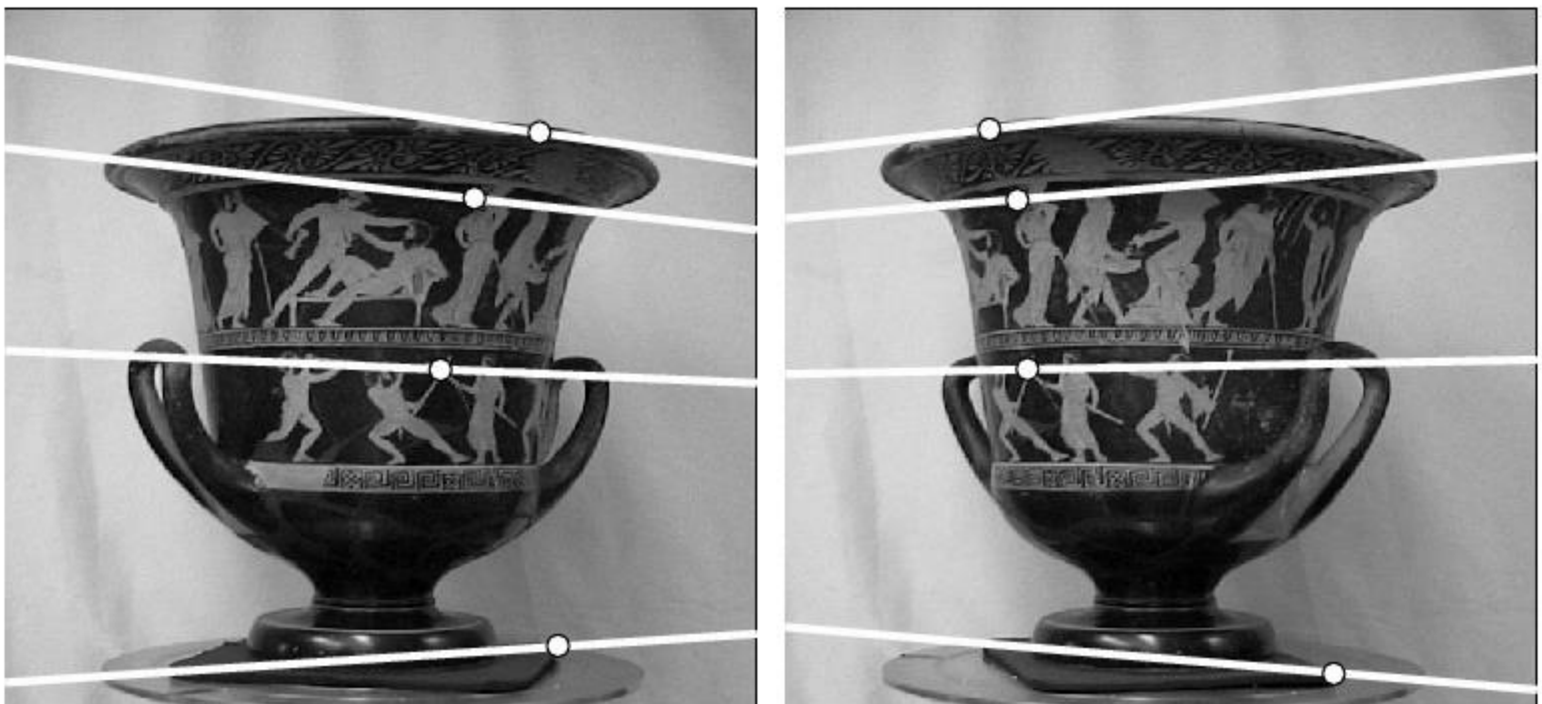
- Output $T^T\, \mathcal{F}\, T'$.

Epipolar geometry example



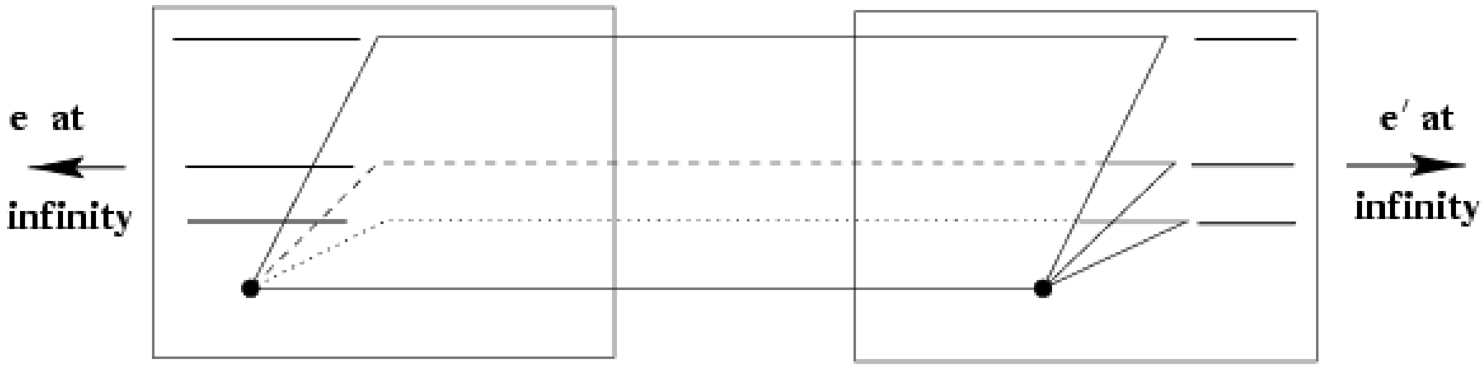
Example: converging cameras

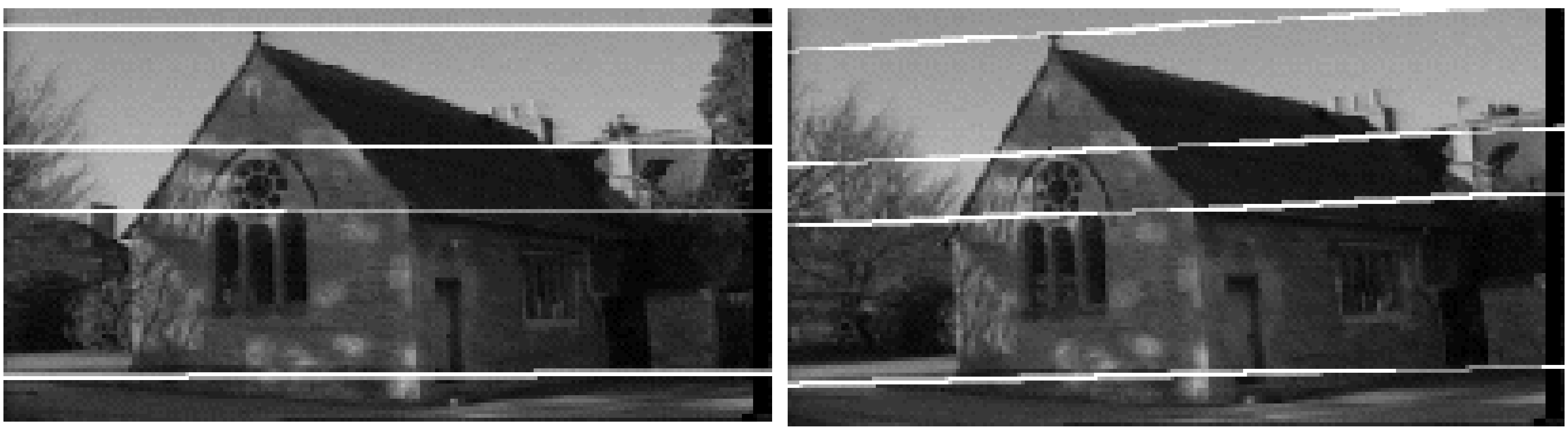




courtesy of Andrew Zisserman

Example: motion parallel with image plane

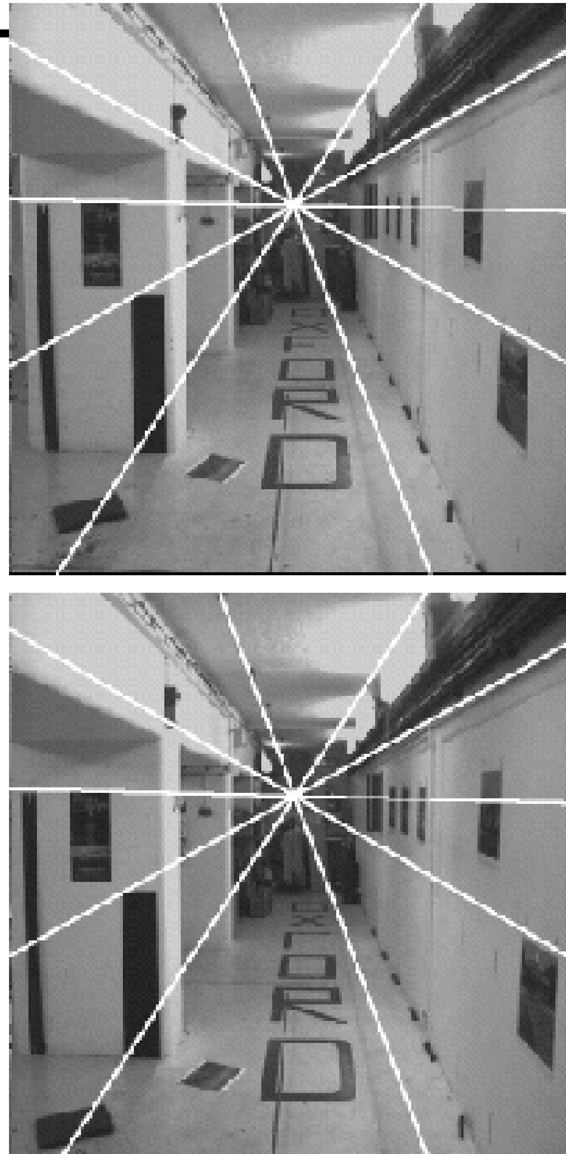


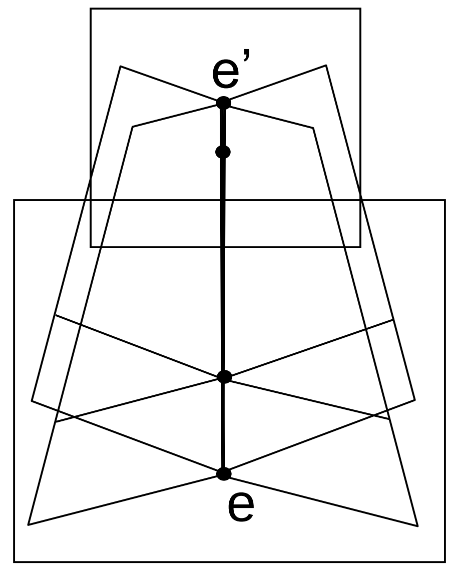


(simple for stereo → rectification)

courtesy of Andrew Zisserman

Example: forward motion

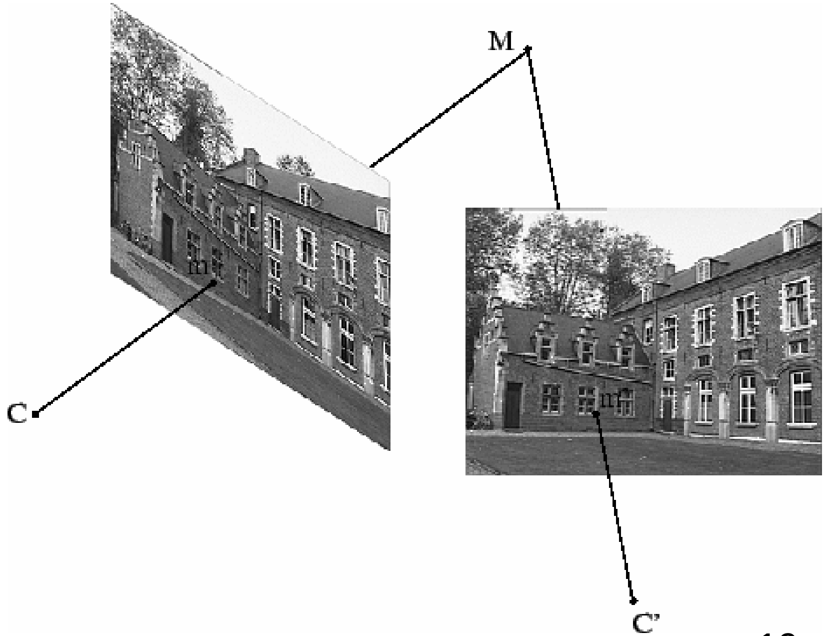




courtesy of Andrew Zisserman

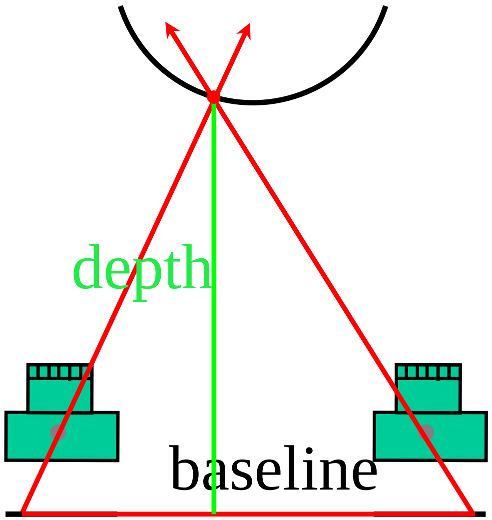
Stereo Reconstruction Problem

- Reconstruction of depth map from a pair of stereo images
- Triangulation
  - 3D point can be obtained as the intersection of the two line of sights



- Requirements
  1. **Relative 3D camera poses and parameters for the stereo cameras (camera calibration)**
  2. **Pixel correspondences (stereo matching)**

Stereo Vision



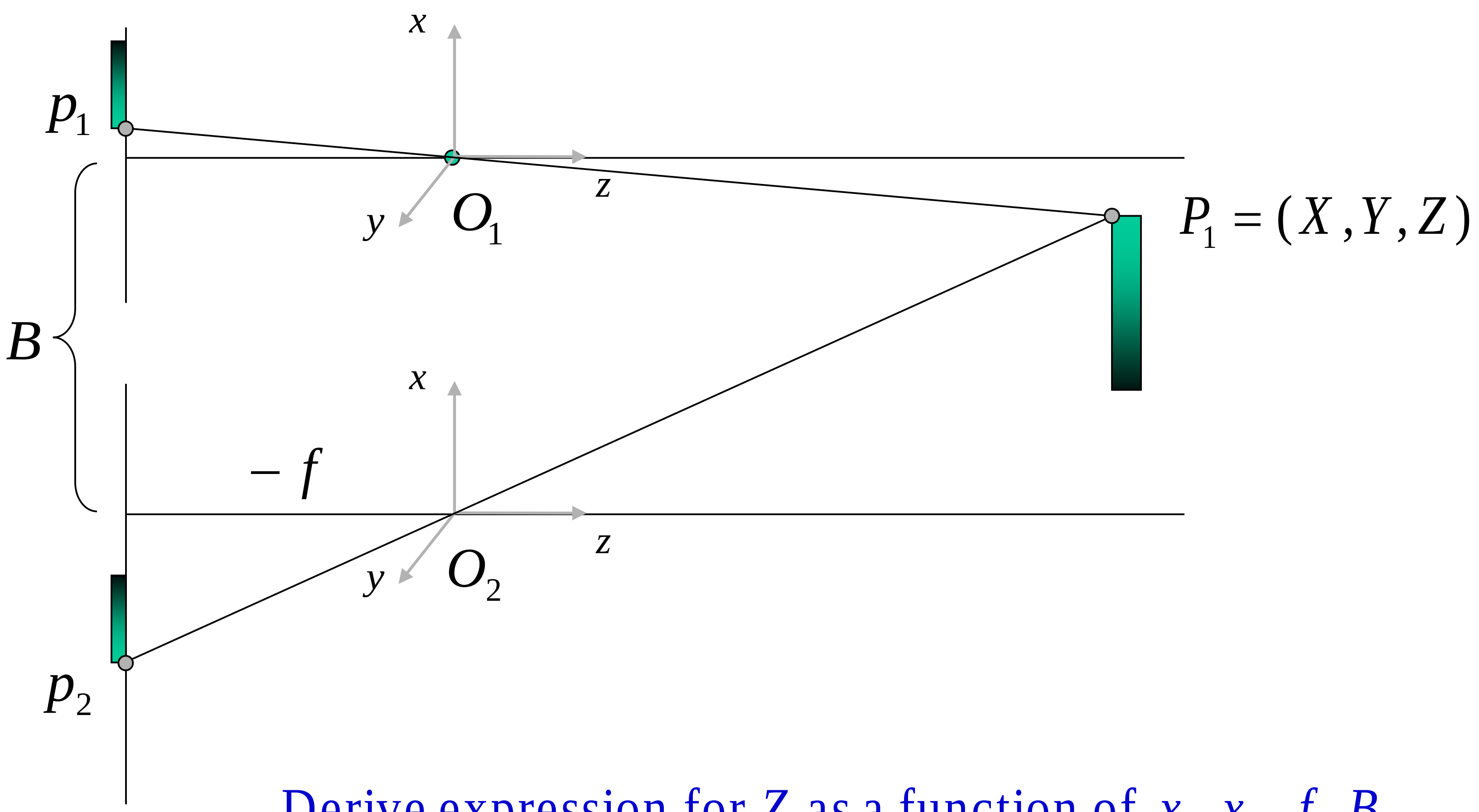
Triangulate on two images of the same point to recover depth.

- Feature matching across views
- Calibrated cameras

Matching correlation windows across scan lines.

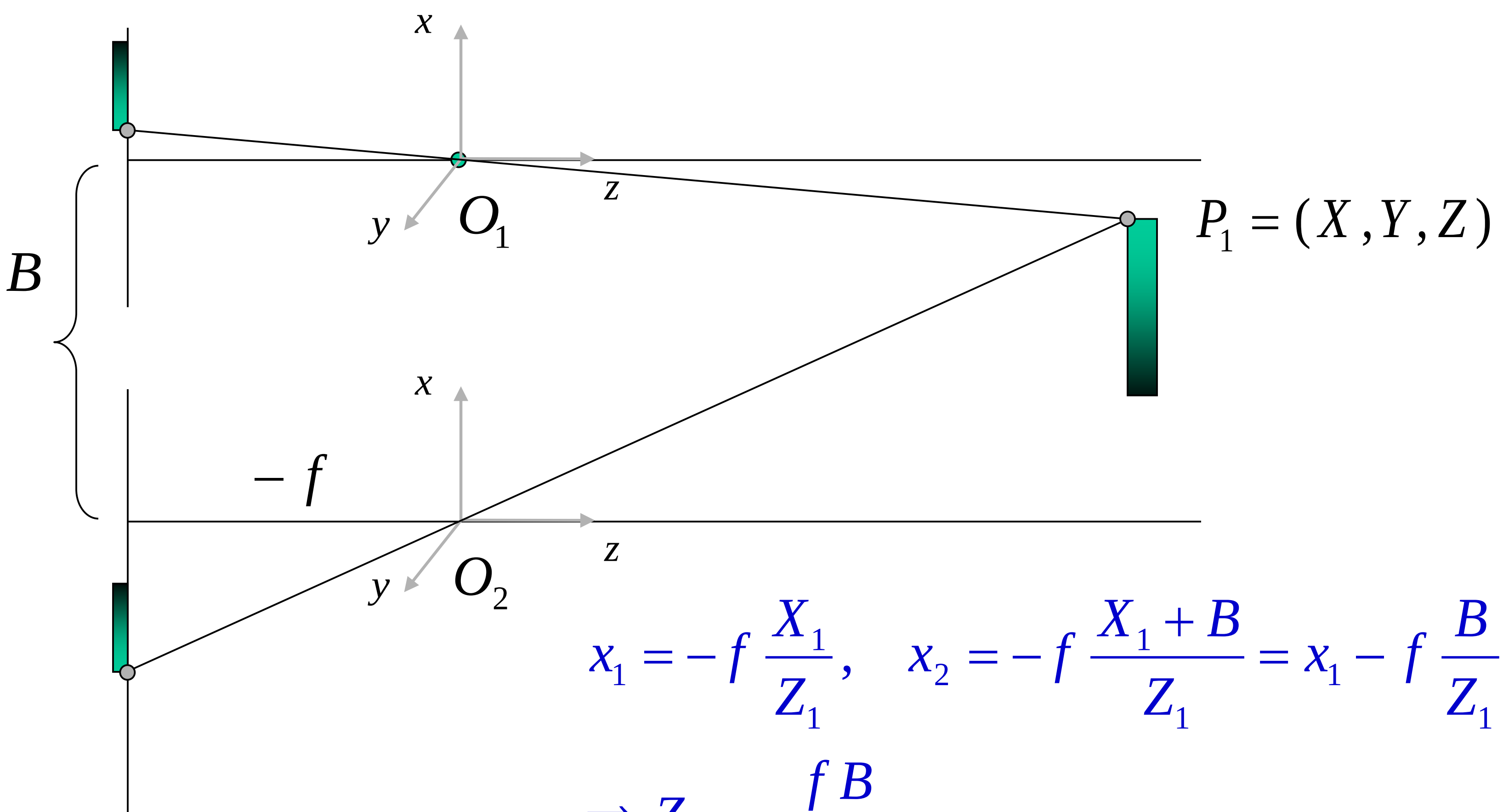
Disparity: deviation between horizontal positions of corresponding points in the calibrated stereo images, directly related to depth.

Basic Stereo Derivation



Derive expression for Z as a function of x_1, x_2, f, B

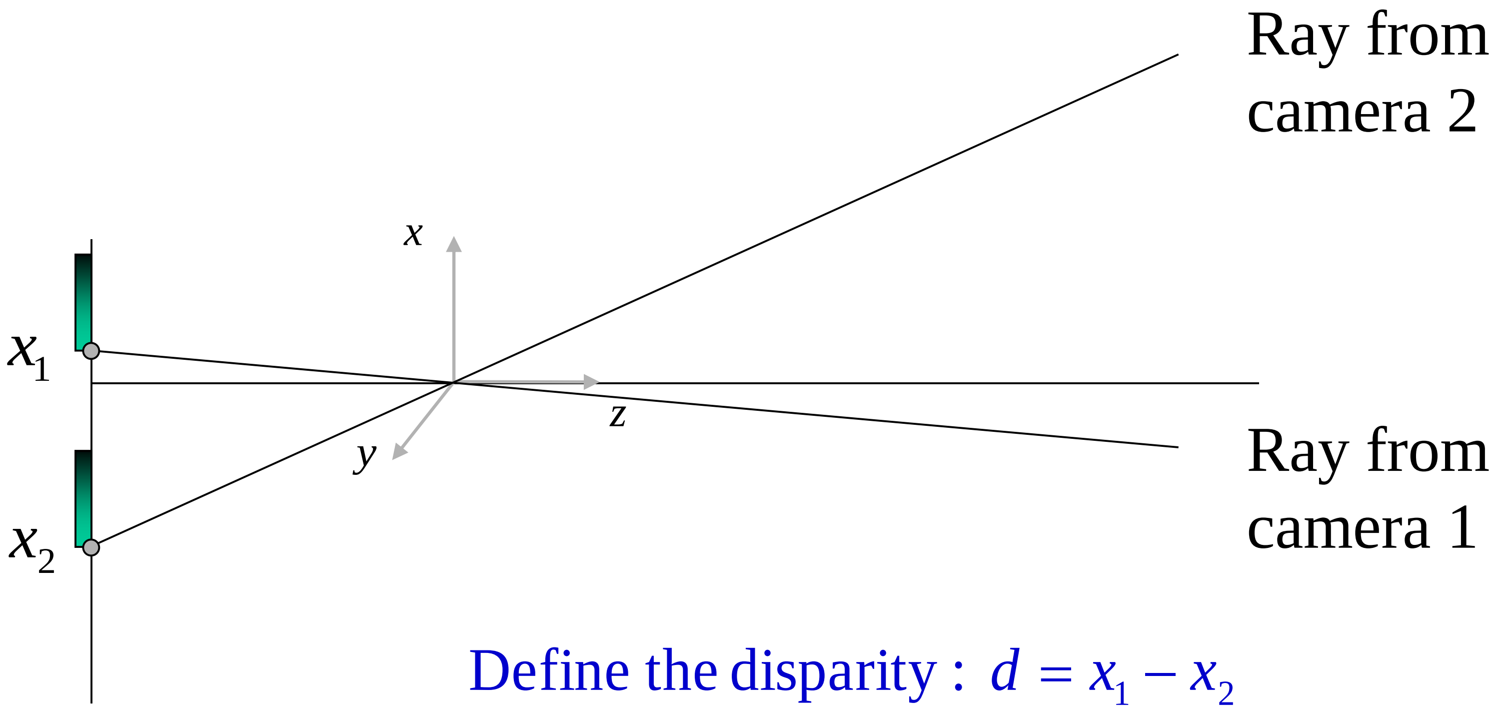
Basic Stereo Derivation



$$x_1 = -f\,\frac{X_1}{Z_1}, \quad x_2 = -f\,\frac{X_1 + B}{Z_1} = x_1 - f\,\frac{B}{Z_1}$$

$$\Rightarrow Z_1 = \frac{f\,B}{x_1 - x_2}$$

Basic Stereo Derivations

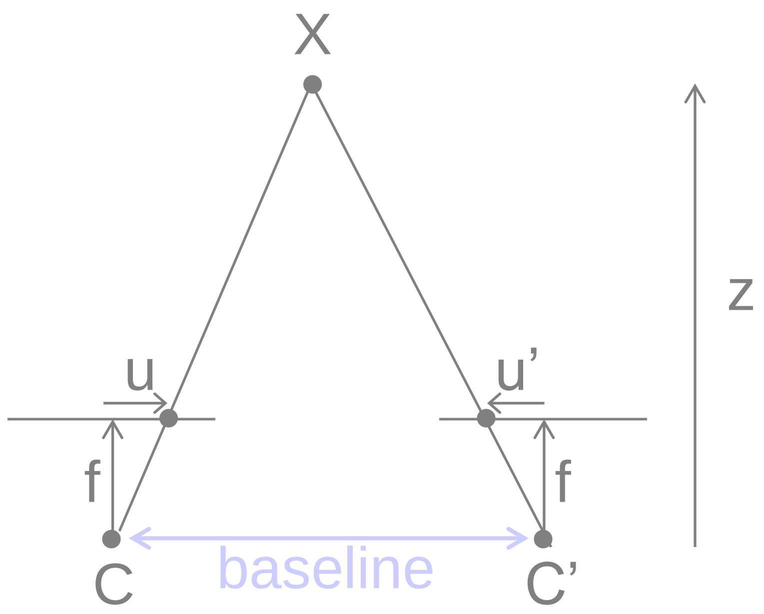


Define the disparity : $d = x_1 - x_2$

$$Z = \frac{f\,B}{d}$$

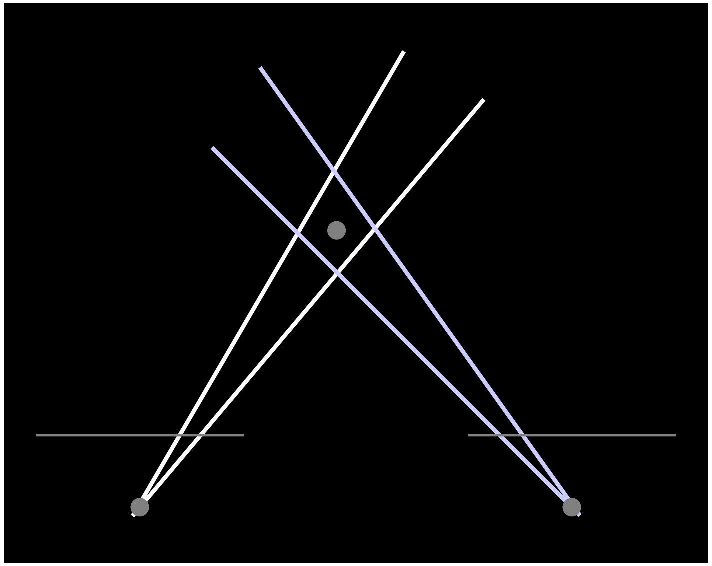
Stereo Reconstruction

- Steps
  - Calibrate cameras
  - Rectify images
  - Compute disparity
  - Estimate depth

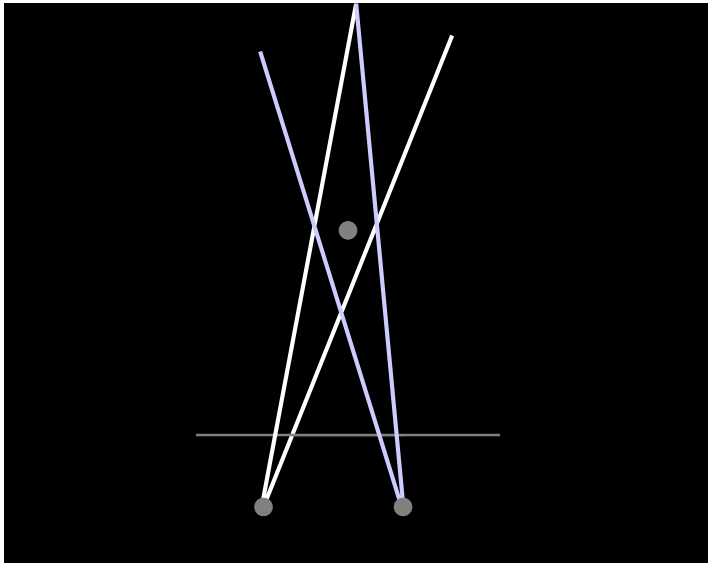


$$disparity = u - u' = \frac{baseline * f}{z}$$

Choosing the Baseline



Large Baseline



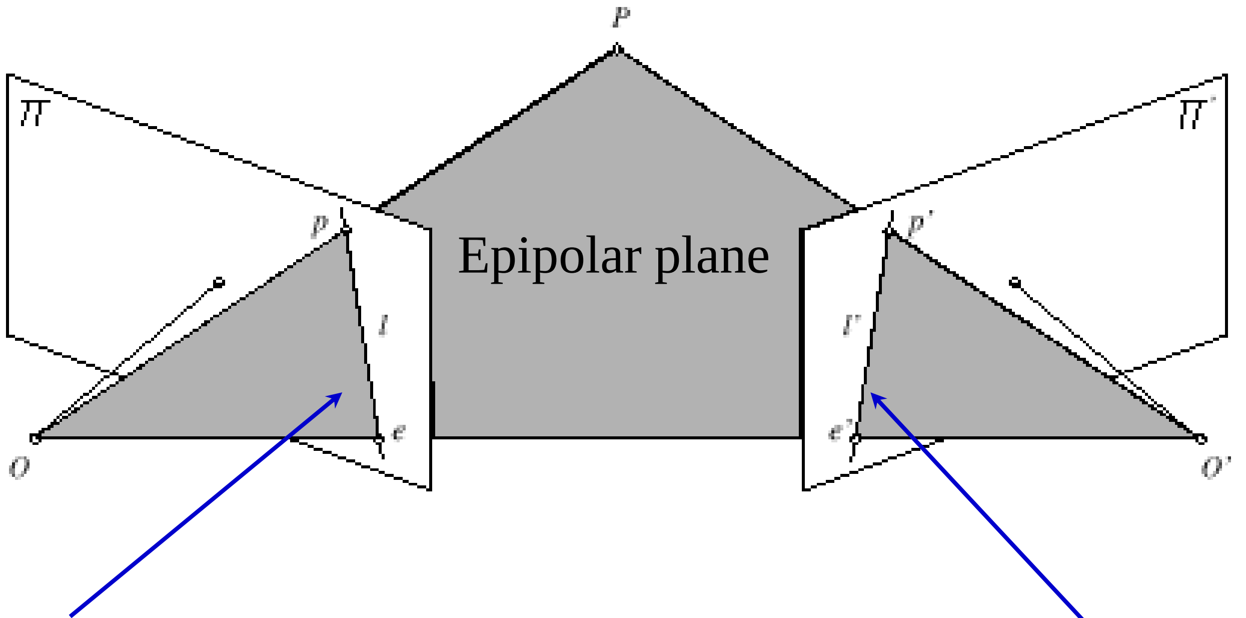
Small Baseline

What's the optimal baseline?

- Too small: large depth error
- Too large: difficult search problem

Epipolar Geometry

- The epipolar geometry is the fundamental constraint in stereo.
- *Rectification* aligns epipolar lines with scanlines



Epipolar line for p'

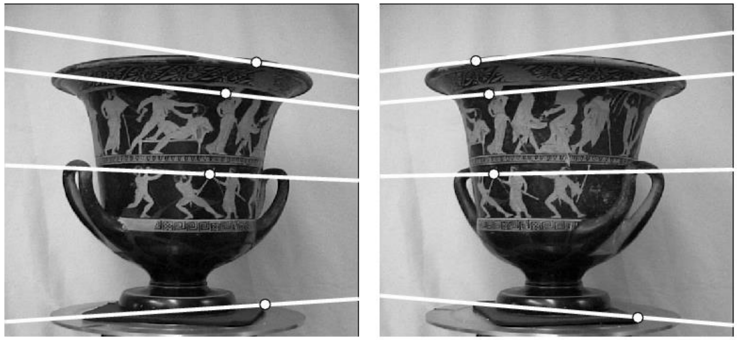
Epipolar line for p

Image rectification

- *Rectification* aligns epipolar lines with scanlines
- Stereo algorithms are often considerably simplified when the images of interest have been rectified.
- Projecting the original pictures onto a common image plane parallel to the baseline joining the two optical centers. The rectified epipolar lines are scanlines of the new images, and they are also parallel to the baseline.
- Given two points p and p' located on the same scanline of the left and right images, with coordinates (u, v) and (u', v). The disparity is defined as the difference $d = u' - u$.

Image Rectification

- Once we have the camera information, the solution space of the matches between two views is restrained from 2D to 1D



- For further simplifying the searching problem, the stereo matching algorithm is performed on the rectified image pair

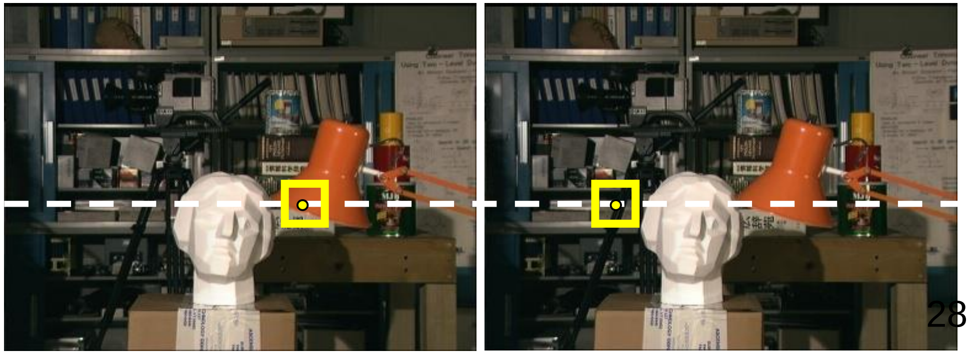
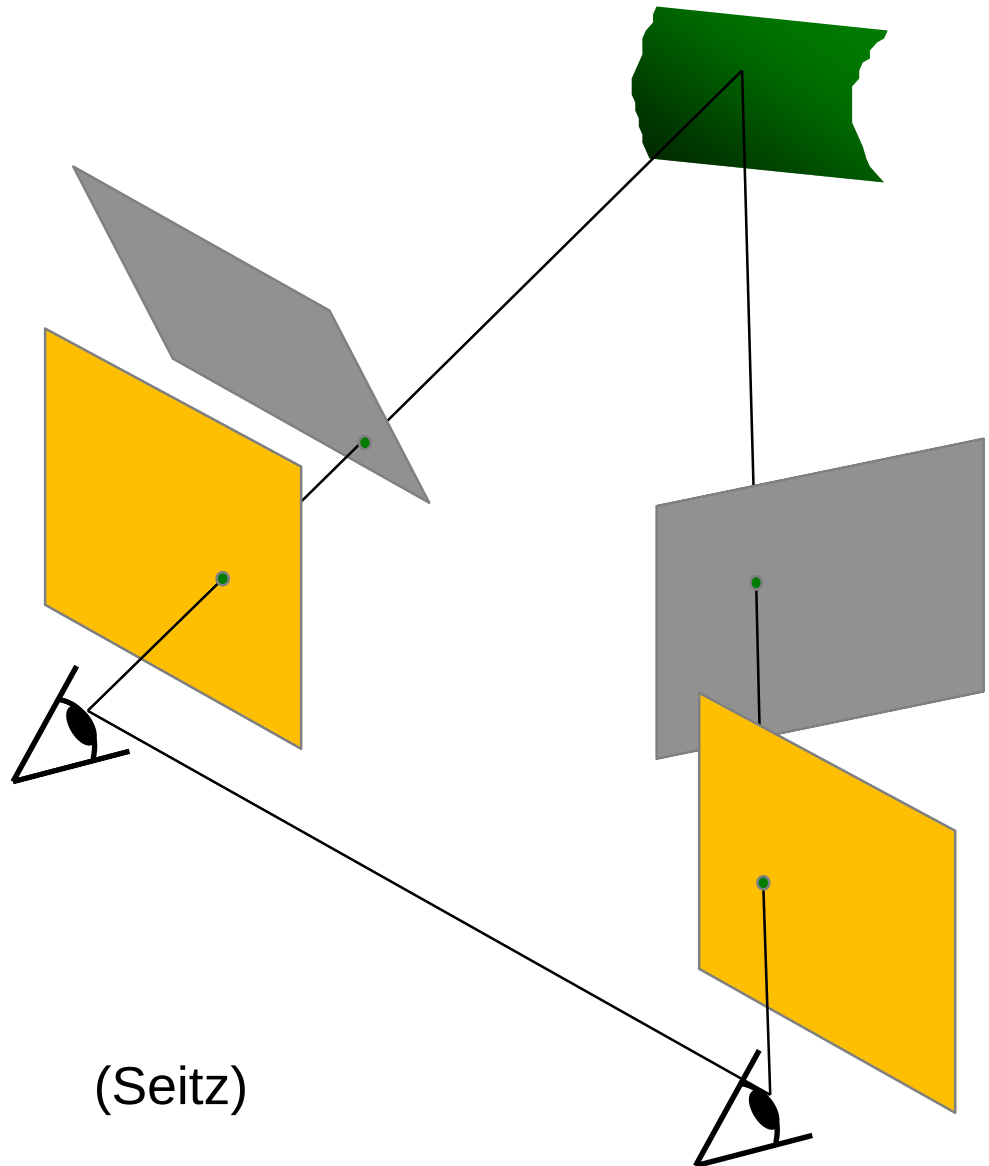


Image rectification

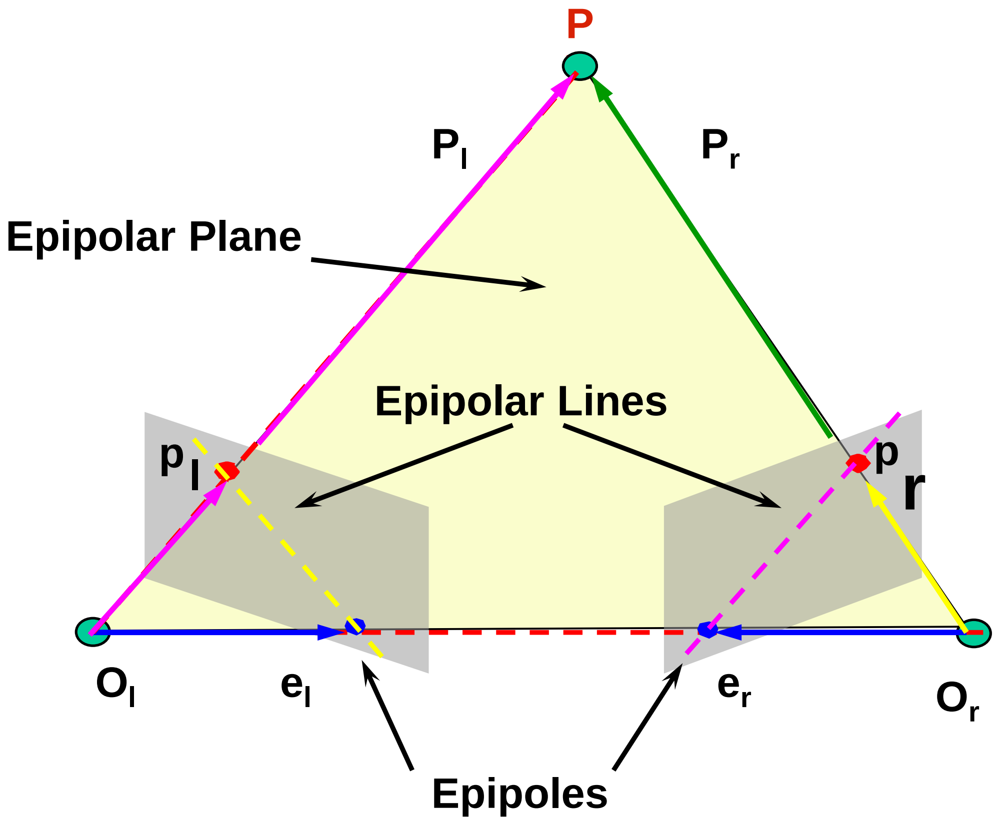
- Given general displacement how to warp the views
- Such that epipolar lines are parallel to each other
- How to warp it back to canonical configuration



(Seitz)

Rectification

- Problem: Epipolar lines not parallel to scan lines



Rectification

- Problem: Epipolar lines not parallel to scan lines

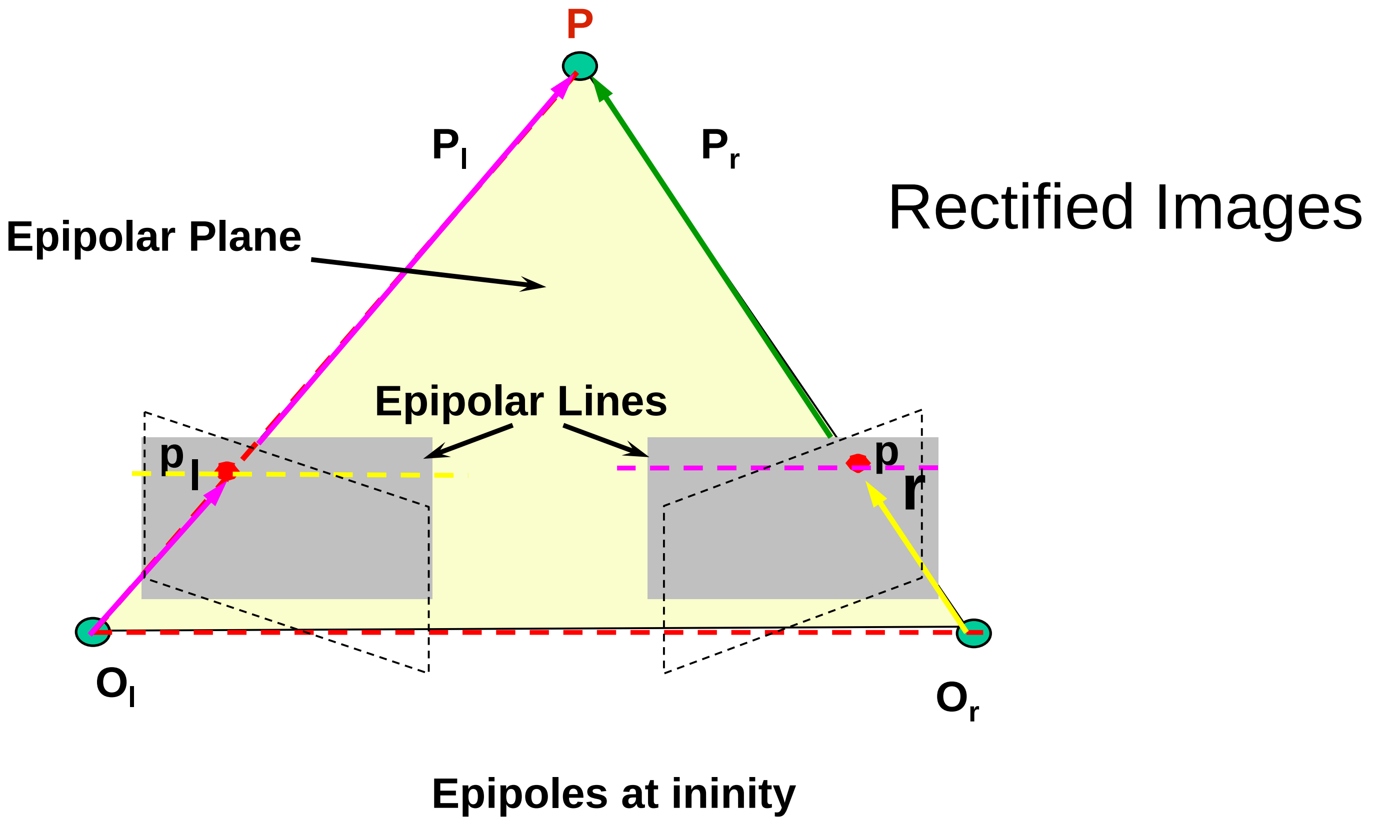
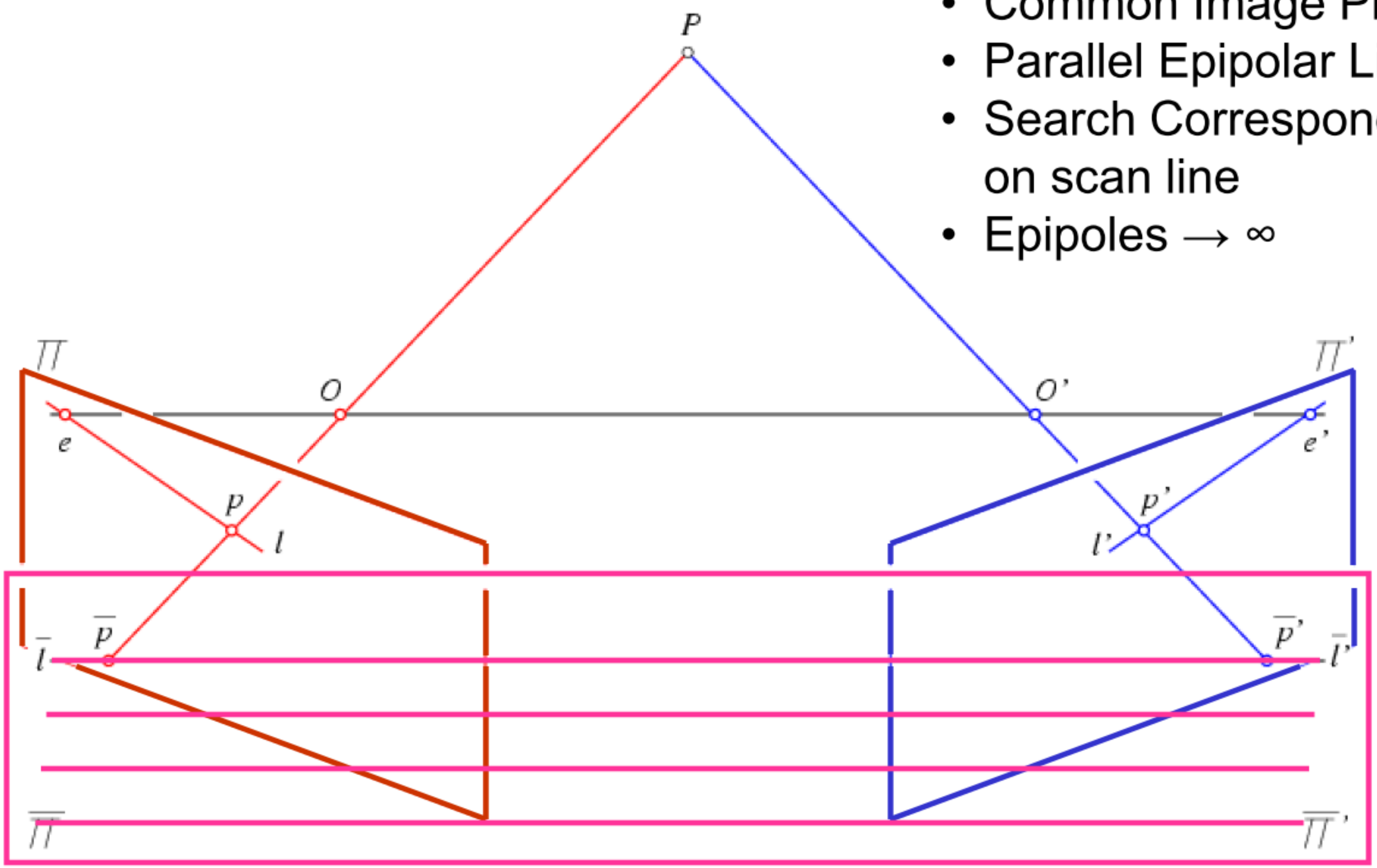


Image Rectification

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

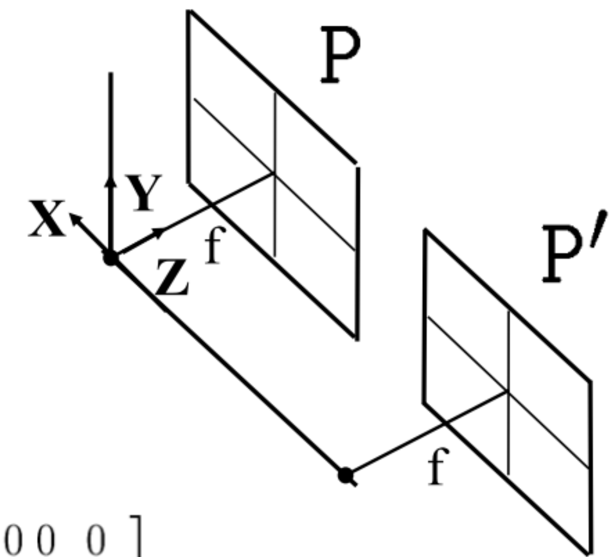


All epipolar lines are parallel in the rectified image plane.

Fundamental matrix for a parallel camera stereo rig

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

$$\mathtt{K} = \mathtt{K}' = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathtt{R} = \mathtt{I} \quad \mathbf{t} = \begin{pmatrix} t_x \\ 0 \\ 0 \end{pmatrix}$$



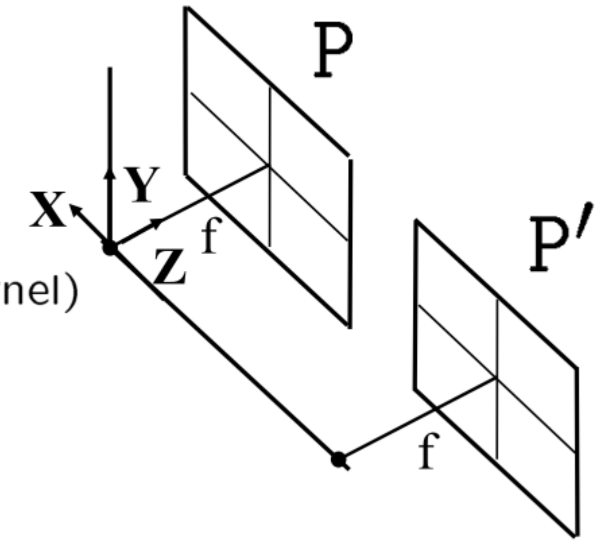
$$\mathbf{F} = \mathtt{K}'^{-\top}[\mathbf{t}]_\times \mathtt{R}\mathtt{K}^{-1}$$

$$= \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix} \begin{bmatrix} 1/f & 0 & 0 \\ 0 & 1/f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

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

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

Case with Parallel camera stereo rig



F is a rank 2 matrix

The epipole $\mathbf{e}$ is the null-space vector (kernel) of F (exercise), i.e. $\mathrm{F}\mathbf{e} = \mathbf{0}$

In this case

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

so that

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

Geometric interpretation ?

Planar Rectification

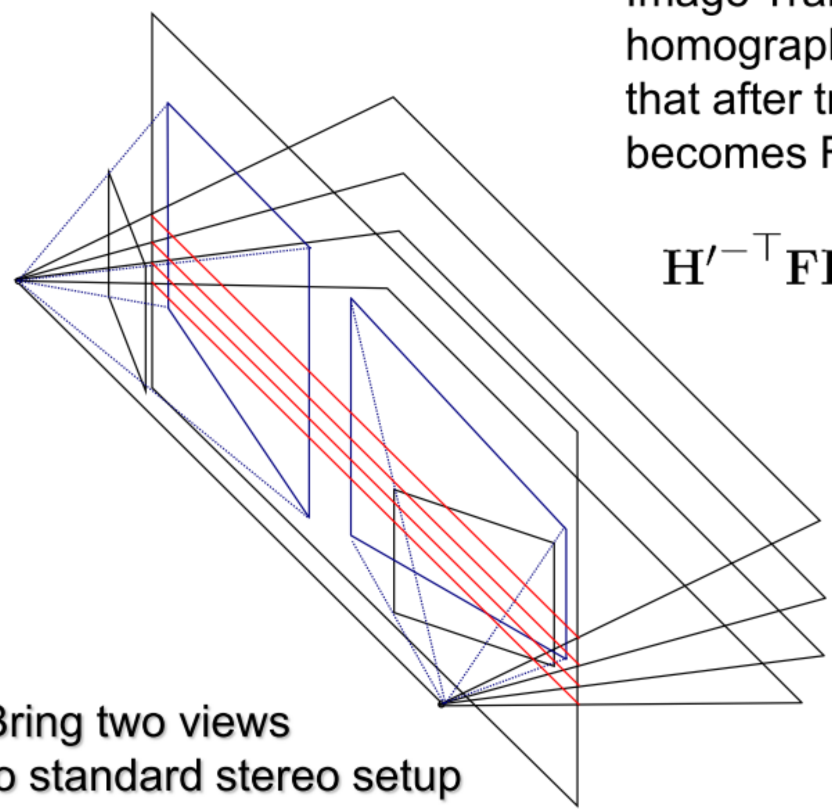


Image Transformations: Find homographies H' and H so that after transformations, F^n becomes F of parallel cameras:

$$\mathbf{H}'^{-\top}\mathbf{F}\mathbf{H}^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

Bring two views to standard stereo setup

(moves epipole to ∞)
(not possible when in/close to image)

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

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

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

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

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:

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

Rectification Algorithm

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, $\mathbf{p}_l = [x, y, f]^\top$ compute

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

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

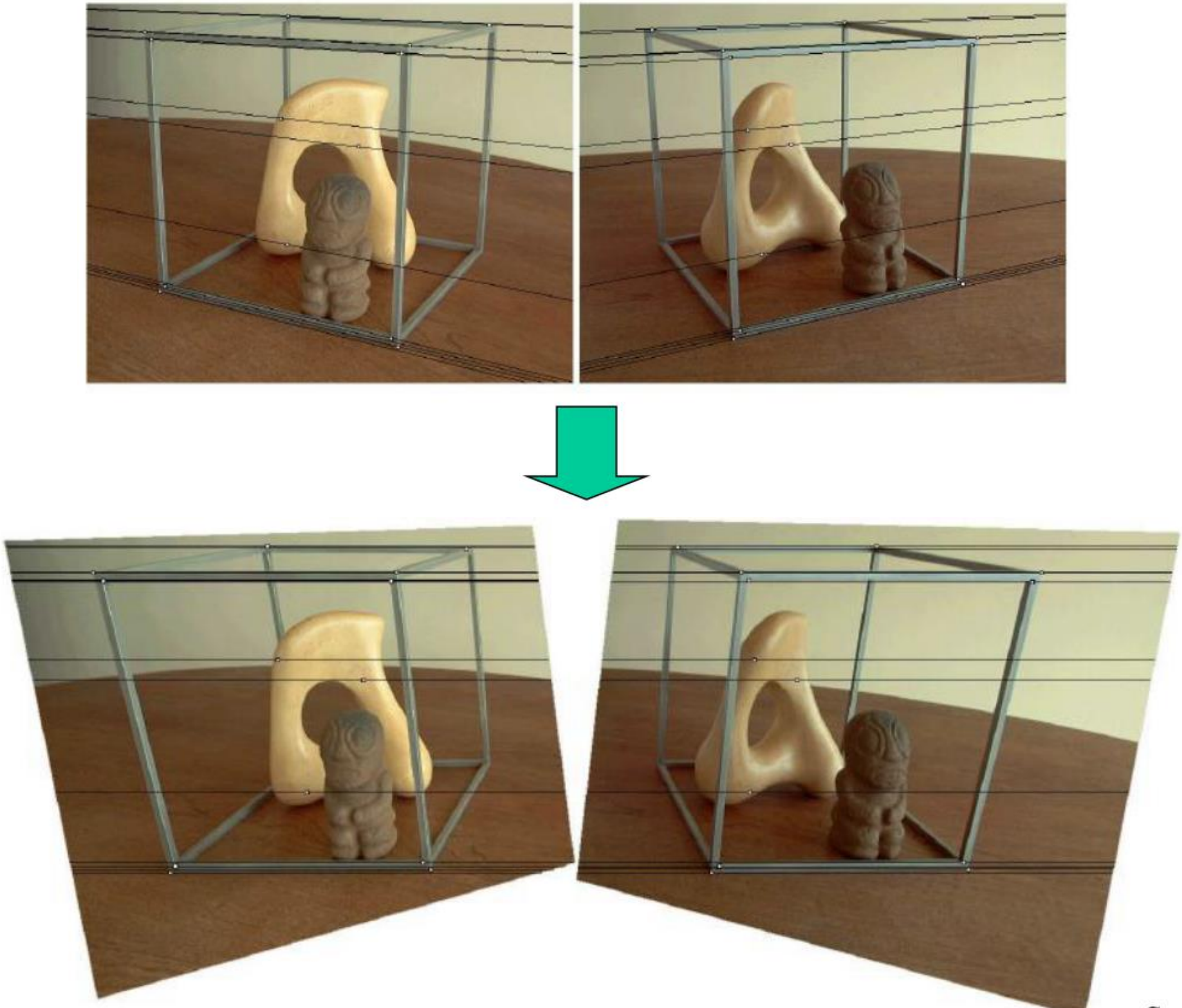
$$\mathbf{p}_l' = \frac{f}{z'}[x', y', z'].$$

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.

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

Epipolar Rectified Images



Source: Alyosha Efros

Stereo Matching Approaches

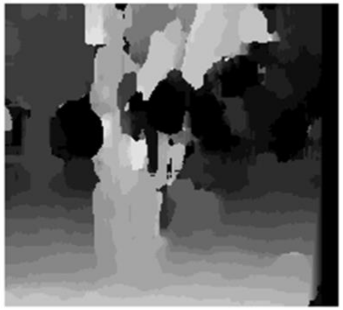
- **Local methods：**
  - ➢ Usually on-line
  - ➢ Key：Window size、Support weight…
  - – Sum of Absolute Difference (SAD) & Sum of Square Difference (SSD)
  - – Color-weighted Correlation
- **Global methods：**
  - ➢ Usually off-line
  - ➢ Key：Energy formulation、Inference algorithms…
  - – Hierarchical Belief Propagation (HBP)
  - – Bitwise + HBP
  - – Symmetric Stereo Matching
  - – Stereo Matching with Segmentation

Local Stereo Matching Methods

- Key：support windows

- Usually on-line
- Advantages
  - Efficient and simple to implement.
  - Can be performed in parallel.

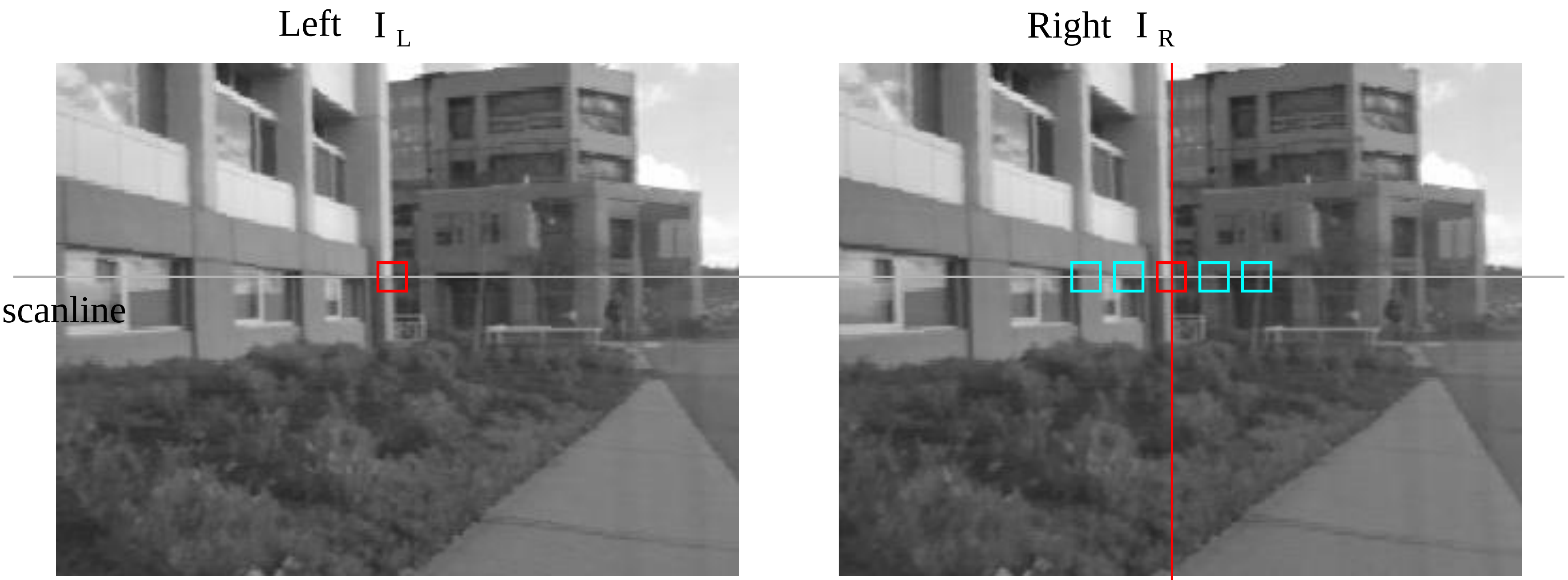
W = 3



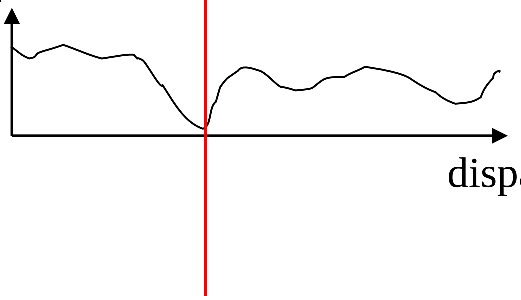
W = 20

- Disadvantages
  - Enforce piecewise smoothness only with local pixels.
  - Poor performance in textureless and occlusion regions.

Correspondence Using Correlation



$$SSD(x,y;d) = \sum_{(u,v)\in W_m(x,y)} [I_L(u,v) - I_R(u-d,v)]^2$$



Correlation

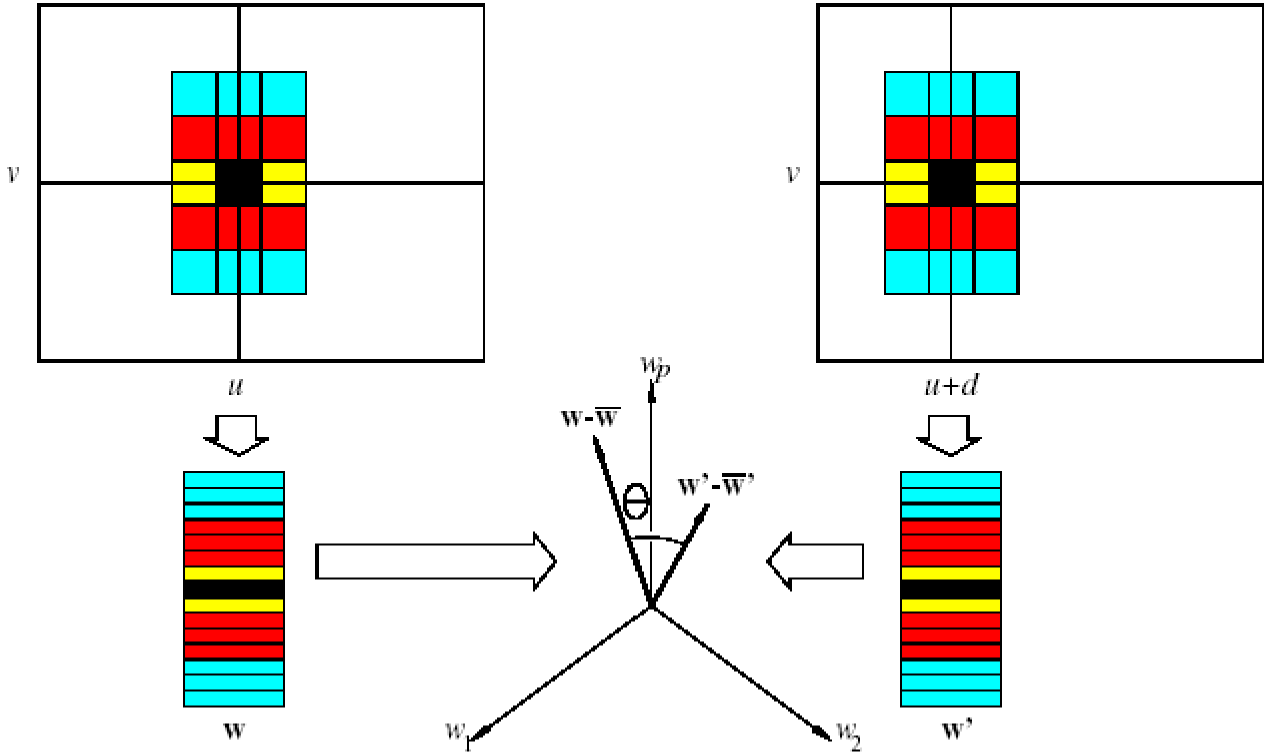


Figure 13.11. Correlation of two 3×5 windows along corresponding epipolar lines. The second window position is separated from the first one by an offset d. The two windows are encoded by vectors $\boldsymbol{w}$ and $\boldsymbol{w}'$ in $\mathbb{R}^{15}$, and the correlation function measures the cosine of the angle θ between the vectors $\boldsymbol{w} - \bar{\boldsymbol{w}}$ and $\boldsymbol{w}' - \bar{\boldsymbol{w}}'$ obtained by substracting from the components of $\boldsymbol{w}$ and $\boldsymbol{w}'$ the average intensity in the corresponding windows.

Correlation

- Define the (normalized) correlation function as:

$$C(d) = \frac{1}{|\, w - \bar{w} \,|} \frac{1}{|\, w' - \bar{w}' \,|} (w - \bar{w}) \cdot (w' - \bar{w}') \qquad \bar{w}, \bar{w}' \text{: average intensity}$$

- The correlation function measures the cosine of the angle θ between the vectors $w - \bar{w}$ and $w' - \bar{w}'$ computed from local windows
- Stereo matches can be found by seeking the maximum of the C function over some pre-determined range of disparities.

Sum of Squared (Pixel) Differences

Left Right

w_L and w_R are corresponding m by m windows of pixels.

We define the window function :

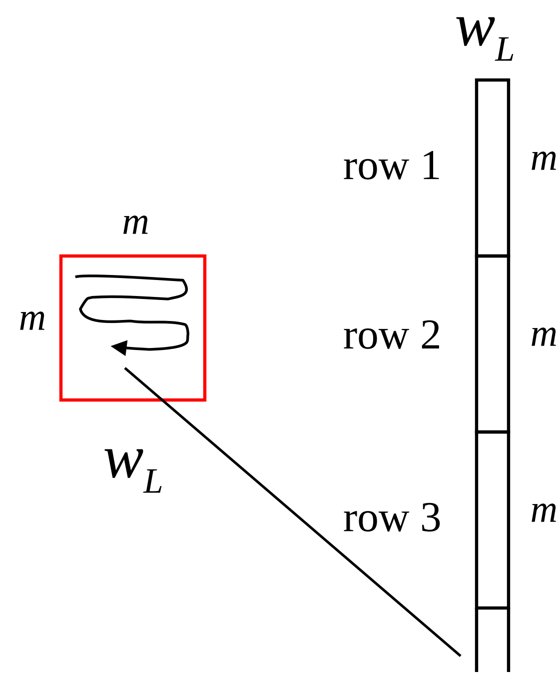
$$W_m(x,y) = \{u, v \mid x - \tfrac{m}{2} \le u \le x + \tfrac{m}{2},\ y - \tfrac{m}{2} \le v \le y + \tfrac{m}{2}\}$$

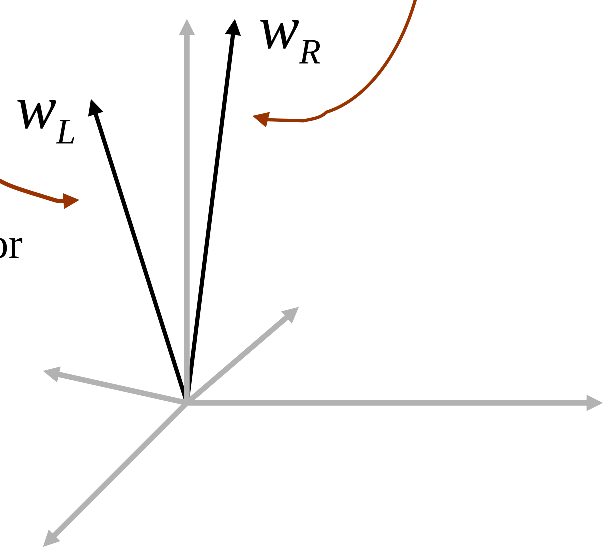
The SSD cost measures the intensity difference as a function of disparity :

$$C_r(x,y,d) = \sum_{(u,v)\in W_m(x,y)} [I_L(u,v) - I_R(u-d,v)]^2$$

Images as Vectors

“Unwrap” image to form vector, using raster scan order





Each window is a vector in an m^2 dimensional vector space. Normalization makes them unit length.

Image Normalization

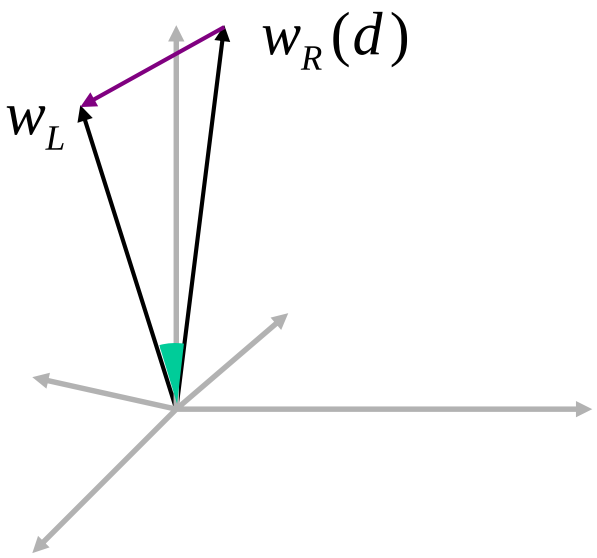
- Even when the cameras are identical models, there can be differences in gain and sensitivity.
- The cameras do not see exactly the same surfaces, so their overall light levels can differ.
- For these reasons and more, it is a good idea to normalize the pixels in each window:

$$\bar{I} = \frac{1}{|W_m(x,y)|} \sum_{(u,v)\in W_m(x,y)} I(u,v)$$ Average pixel

$$\|I\|_{W_m(x,y)} = \sqrt{\sum_{(u,v)\in W_m(x,y)} [I(u,v)]^2}$$ Window magnitude

$$\hat{I}(x,y) = \frac{I(x,y) - \bar{I}}{\|I - \bar{I}\|_{W_m(x,y)}}$$ Normalized pixel

Image Metrics



(Normalized) Sum of Squared Differences

$$C_{\text{SSD}}(d) = \sum_{(u,v)\in W_m(x,y)} [\hat{I}_L(u,v) - \hat{I}_R(u-d,v)]^2$$

$$= \left\| w_L - w_R(d) \right\|^2$$

Normalized Correlation

$$C_{\text{NC}}(d) = \sum_{(u,v)\in W_m(x,y)} \hat{I}_L(u,v)\hat{I}_R(u-d,v)$$

$$= w_L \cdot w_R(d) = \cos\theta$$

$$d^* = \arg\min_d \left\| w_L - w_R(d) \right\|^2 = \arg\max_d \, w_L \cdot w_R(d)$$

Global Approaches to Stereo Matching

Graph Cuts

Belief Propagation

Boykov et al., Fast Approximate Energy Minimization via Graph Cuts,
International Conference on Computer Vision, September 1999.

Stereo Matching Using Belief Propagation
Jian Sun, Heung-Yeung Shum, and Nan-Ning Zheng
ECCV 2002

Global Stereo Matching Methods

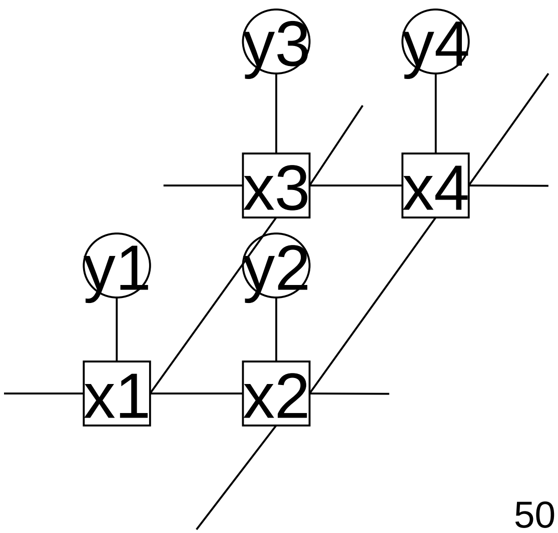
- Key：Cost function
- Usually off-line
- Advantages
  - More accurate.
  - Better estimation for textureless regions.
- Disadvantages
  - Often combined with other techniques to improve the results, but this also increases computation cost.

Belief Propagation

Graph Cuts

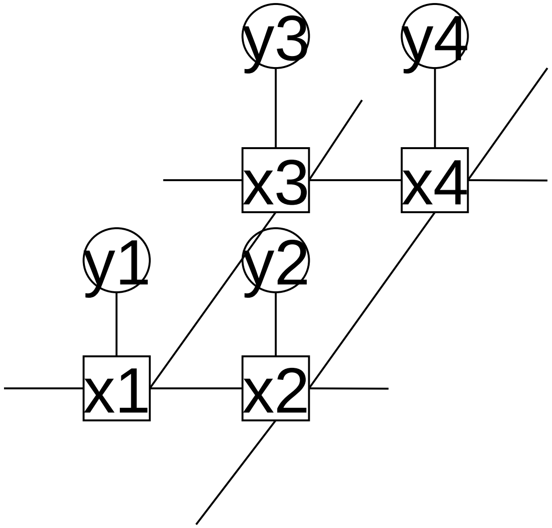
Markov Random Field (MRF)

- Markov Random Field (MRF)
  - MRFs are a kind of statistical model
  - Easily describe local relationship
  - MRFs replace temporal dependency of Markov chains with spatial dependency



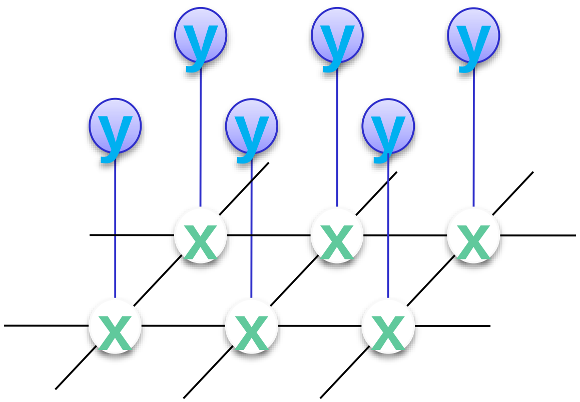
Markov Random Field (MRF)

- The undirected graph which is often used in many vision problems
  - Observation node: y
    - Represent some visible information in low-level problem
  - Hidden node: x
    - Each node own a corresponding observation node *y*



Markov Random Field (MRF)

MRF Model：



x：Hidden node
y：Observation node

Energy Formulation：

$$\boxed{E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} V(f_p, f_q)}$$

Data term　　Smoothness term

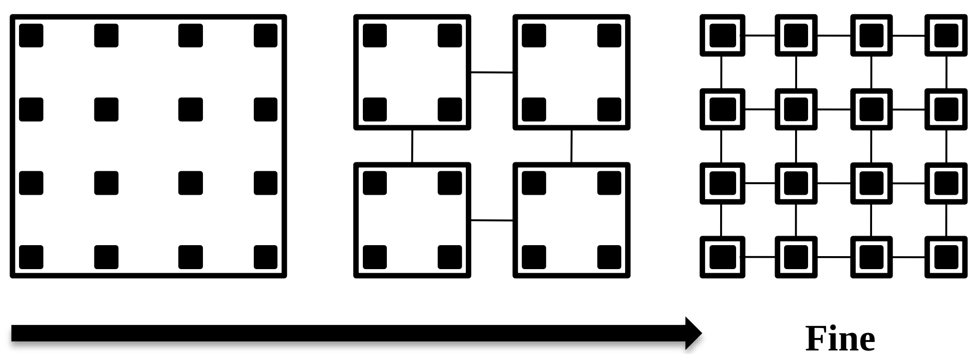
$$\boxed{D_p(f_p) = \frac{1}{3} \sum_{c \in \{R,G,B\}} \left| I_c(p) - I_c(p - f_p) \right|}$$

$$\boxed{V(f_p, f_q) = \min(\left| f_p - f_q \right|, d)}$$

Hierarchical Structure：

…

…



Coarse **Fine**

Belief Propagation (BP)

- An efficient iterative algorithm for getting approximate inference the values of hidden node
- Two main steps :
  - (1) Simplified Graph Construction
  - (2) Message Passing until convergence
- While loops exists, BP may not converge

Local Message Passing for Trees

- Sum-product algorithm

$$m_{ij}(x_j) \leftarrow \alpha \sum_{x_i} \psi_{ij}(x_i, x_j)\phi_i(x_i) \prod_{x_k \in \mathcal{N}(x_i)\backslash x_j} m_{ki}(x_i)$$

$$b_i(x_i) \leftarrow \alpha \phi_i(x_i) \prod_{x_j \in \mathcal{N}(x_i)} m_{ji}(x_i)$$

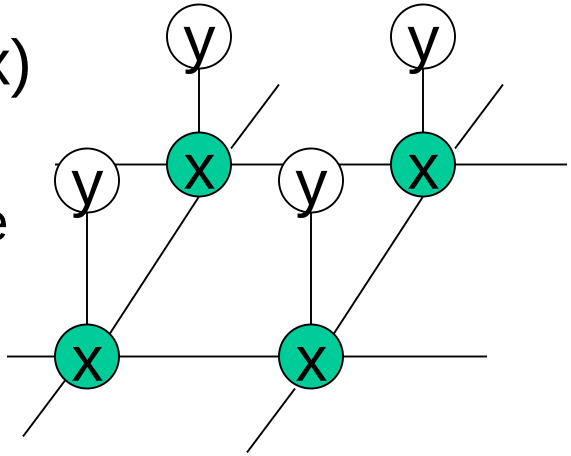
  – Find marginal
- Max-product algorithm
  – Maximum a posteriori (MAP) probabilities

$$m_{ij}(x_j) \leftarrow \alpha \max_{x_i} \psi_{ij}(x_i, x_j)\phi_i(x_i) \prod_{x_k \in \mathcal{N}(x_i)\backslash x_j} m_{ki}(x_i)$$

  – Find a setting of the variables corresponding to the largest probability

Loopy Belief Propagation

- Murphy et al. give a new algorithm which can be implemented on the graphs with loops
- Convergence not guaranteed, but seems to work well in practice
- Message passing
  - Observation potential function (y-x)
    - To Describe visual information
    - Ex: pattern match, intensity difference
  - Neighbor likelihood (x-x)
    - Relationship between hidden nodes
  - Message update
    - Combine observation and neighborhood



K. P. Murphy, Y. Weiss, and M. I. Jordan. "Loopy belief propagation for approximate inference: an empirical study." In Proc. of UAI, pp. 467-475, 1999.

Loopy Belief Propagation (LBP)

- Loopy Belief Propagation
  - Iterative algorithm

- Discontinuity term (neighborhood)

$$V(f_p - f_q) = \min(|f_p - f_q|, d)$$

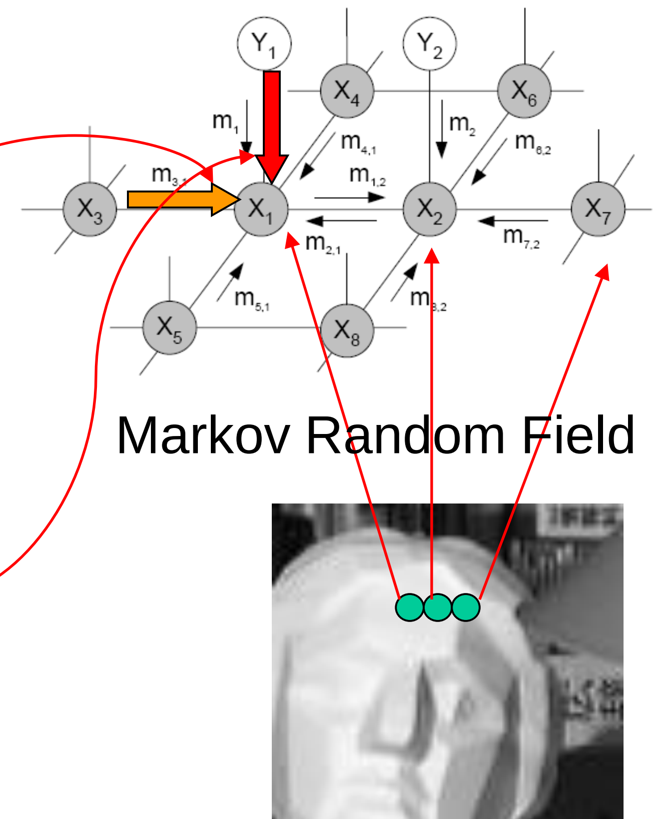
λ

  - f : disparity label
  - d : truncation term

- Data term (observation term)

$$D_p(f_p) = \lambda \min(|I_l(x, y) - I_r(x - f_p, y)|, \tau)$$

  - τ : truncation term
  - : scaling factor



Markov Random Field

Efficient Belief Propagation for Early Vision in CVPR 2004
Pedro F. Felzenszwalb and Daniel P. Huttenlocher

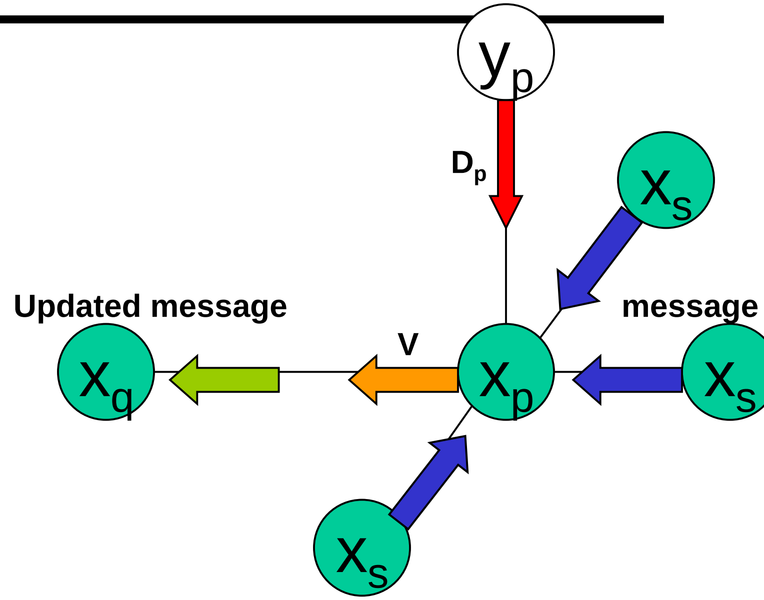
LBP (cont.)

- At each iteration
  - Message update (node p to node q)

t-th iteration

$$m^{t}_{p\to q}(f_q)=\min_{f_p}\Big(\overbrace{V(f_p-f_q)}^{\text{Discontinuity term}}+h(f_p)\Big)$$

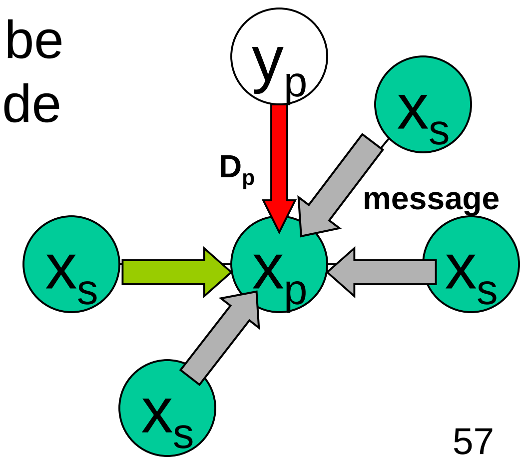
$$\text{where } h(f_p)=\underbrace{D_p(f_p)}_{\text{Data term}}+\underbrace{\sum_{s\in N(p)/q} m^{t-1}_{s\to p}(f_p)}_{\text{Neighborhood messages at the last iteration}}$$



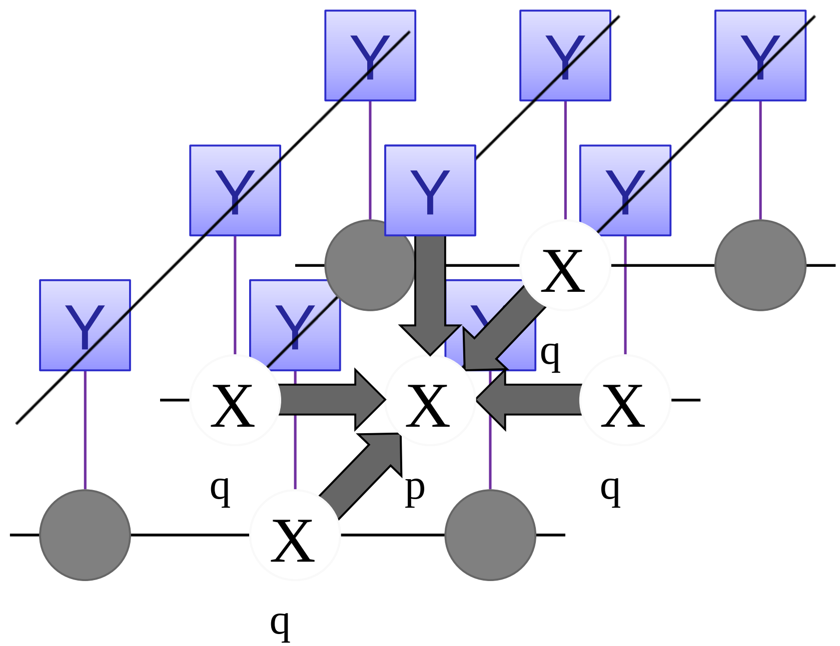
- After enough iterations, the energy of the graph will be converged. And the label lets the energy of each node minimized, assign it to be the disparity of the pixel.

$$D_{belief}(f_p)=\min_{f_p}\left(D_p(f_p)+\sum_{s\in N(p)} m^{t}_{s\to p}(f_p)\right)$$

f_p : disparity level of p coordinate



Belief Propagation (BP)



MRF structure

Y : **Observation node**.
Represent some visible information

X : **Hidden node**.
Each node own a corresponding observation node Y.

- Energy function:
- At each iteration, messages are computed.
- After T iterations, a belief vector is computed for each node

$$E(f) = \sum_{p \in P} D_p(f_p) + \sum_{(p,q) \in N} V(f_p, f_q)$$

$$m^t_{pq}(f_q) = \min_{f_p} \left(D_p(f_p) + V(f_p, f_q) + \sum_{s \in N(p) \backslash q} m^{t-1}_{sp}(f_p) \right)$$

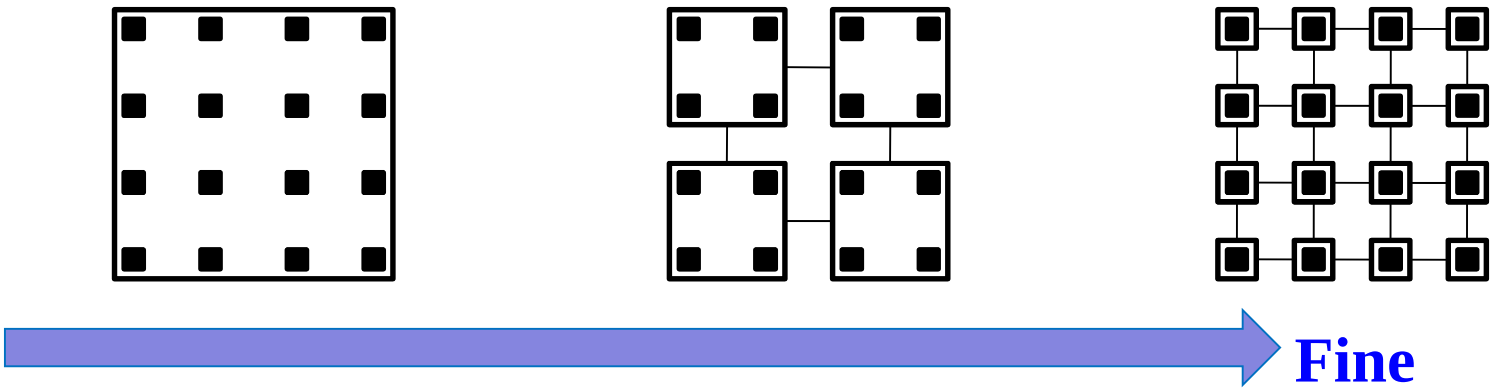
$$b_q(f_q) = D_q(f_q) + \sum_{p \in N(q)} m^T_{pq}(f_q)$$

Data term Smoothness term

Hierarchical Belief Propagation (HBP)

…

…



Coars
e

Fine

…

…

SSD

Binomial filter

Regular diffusion

HBP

Image size: 384x288

Disparity level: 16

Method	Time (s)
SSD	0.157
Binomial filter	5.922
Diffusion	3.969
HBP	2.140

Summary

- Epipolar geometry
- Fundamental matrix estimation
  - Normalized 8-pont algorithm
- Stereo vision
- Stereo image rectification
- Stereo image matching
  - Local method
  - Global method