VAT Workshop on

Computer vision, Graphics and Image processing

# Class 31 and 32: Single view, stereo and Multiview reconstruction


Dr. Uma Mudenagudi

Professor,

Department of Electronics and Communication,

BVB College of Engineering and Technology, Hubli


Class-31, 32 30-06-2010 10.00-1.15pm

# Outline

- Introduction
- Single view reconstruction
  - Camera Calibration using homographies
  - Computing 3D coordinates
  - Results
- Stereo reconstruction
  - Introduction
  - Stereo reconstruction and results
- Multiview reconstruction
  - Introduction
  - Multi view reconstruction and results
- Summary and Conclusions

# Methods of reconstruction/depth estimatio

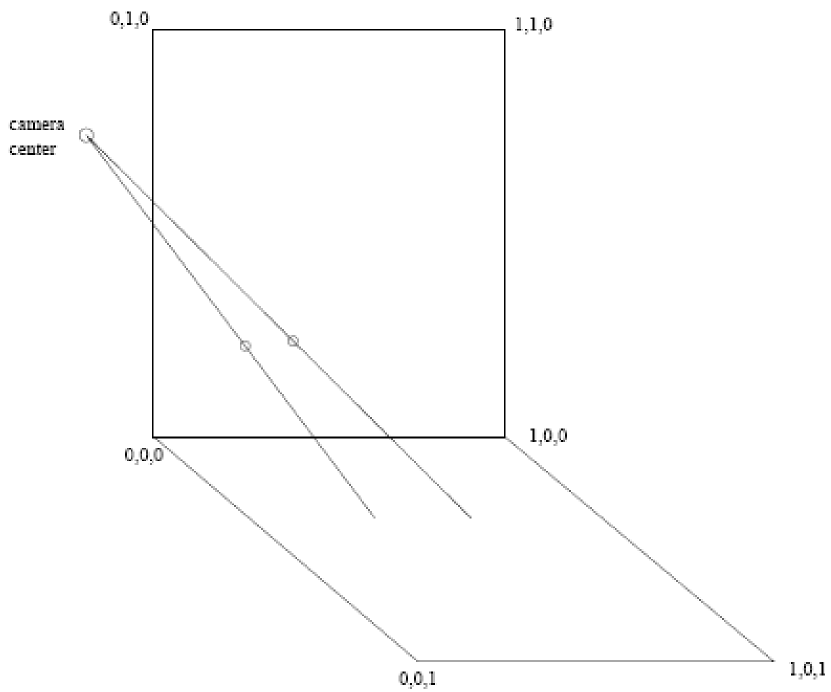
- Process of capturing the depth and appearance of real scene from image/s.
- Methods:
  - Stereo: dense correspondence
  - Structure from motion
  - Depth from focus: needs more photographs
  - Depth from defocus: needs careful settings of the camera
  - Depth from multiple views

# Single view $3D$ reconstruction

- http://make3d.cs.cornell.edu/code.html
- Given a single image of a 3D scene, the aim is to generate a partial 3D model by manually registering two world planes.
- This is called interactive registration.
- The method is based on the computation of homographies $H$ and $G$ from two world planes to the image.
- Homography: The mapping from plane to plane

# Single view $3D$ reconstruction

- The basic case



# Camera calibration

- Let $H$ and $G$ be the homographies from the world $X-Z$ and $X-Y$ planes to the image respectively.
- The two crucial properties of $H$ and $G$ are:
  - *The third column of both the homographies are equal(up to scale)*: world origin (0,0,0) has same coord (0,0) in both.
  - *The first column of both the homographies are equal(again up to scale)*: images of the points at infinity along $X$ direction.
- $8+8=16$ unknowns reduce to 11 (8+3 in the second)

# Camera calibration contd..

- The $3 \times 4$ projection matrix of the camera can be read out from the columns of the homographies

$$H = [H(1), H(2), H(3)] \text{ and } G = [G(1), G(2), G(3)]$$

- The projection matrix $P$ for the camera is

$$P = [H(1)\ \ G(2)\ \ H(2)\ \ H(3)]$$

first three col: vanishing points in the respective directions and last col:projection of world origin.

- The $P$-matrix obtained above can be written as $K[R|t]$ where $3 \times 3$ matrix $K$ is the matrix of camera internals, $R$ is the $3 \times 3$ rotation matrix from the world coordinate system to camera coordinate system and -$R^t t$ are the coordinates of the camera center.

# Camera calibration contd..

- The first $3 \times 3$ of $P$ is $KR$. Letting $\widetilde{P} = KR$.
- Clearly

$$\widetilde{P} * \widetilde{P}^t = KK^t$$

- The camera internal matrix has the general form

$$K = \begin{pmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Hence $KK^t$ will have the form

$$X = KK^t = \begin{pmatrix} \alpha_u^2 + u_0^2 + s^2 & \alpha_v s + u_0 v_0 & u_0 \\ \alpha_v s + u_0 v_0 & \alpha_v^2 + v_0^2 & v_0 \\ u_0 & v_0 & 1 \end{pmatrix}$$

# Calibration contd..

- Thus the camera internals can be directly obtained from the above matrix.
- Normalising **X** and making $X_{33}$ equal to 1.
- Now,

$$u_0 = X_{31} v_0 = X_{32} \alpha_v = \sqrt{X_{22} - v_0^2} s = \frac{X_{21} - u_0 v_0}{\alpha_v} \alpha_u = \sqrt{X_{11} - s^2 -$$

# Calibration contd..

- Once **K** has been obtained, **R** and **t** can be obtained as

$$\mathbf{R} = \mathbf{K}^{-1} * \widetilde{\mathbf{P}} \quad \text{where } \widetilde{\mathbf{P}} \text{ is } \mathbf{KR}.$$

$$\mathbf{t} = \mathbf{K}^{-1} * \mathbf{Kt} \quad \text{where } \mathbf{Kt} \text{ is the last column of } \mathbf{P}.$$

- Once **R** and **t** are computed, the camera center can be computed as -$R^t$**t**.
- Alternately, the camera center can be directly computed from the homographies **H** and **G** as follows:

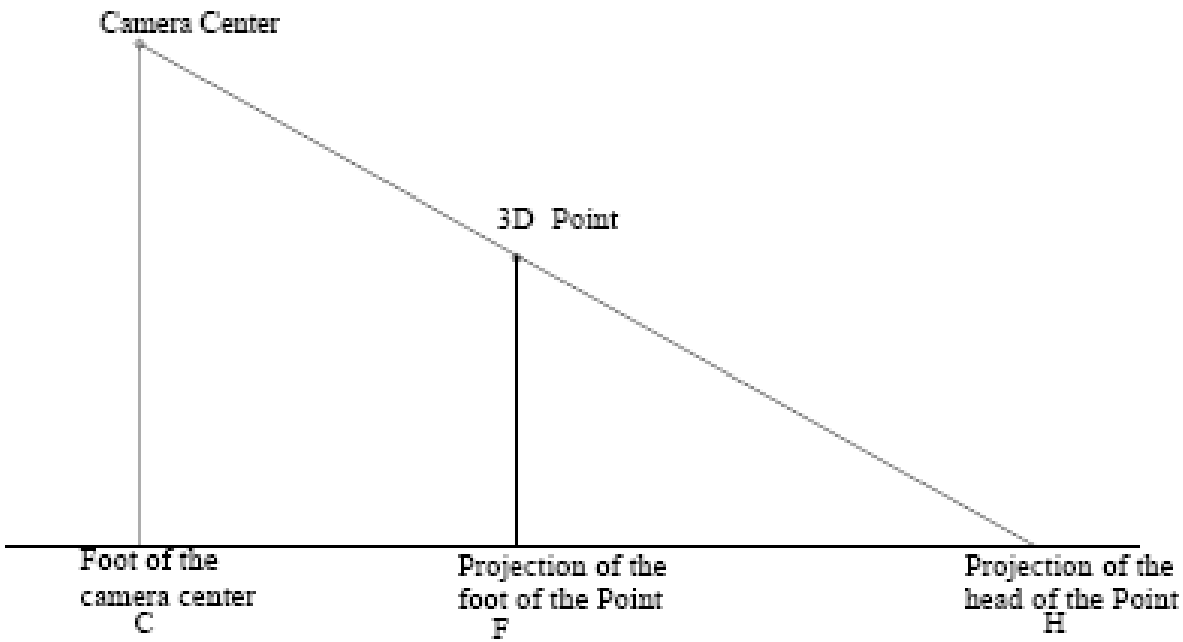
$$T = G^{-1}H = \lambda \begin{pmatrix} -C_z & C_x & 0 \\ 0 & C_y & 0 \\ 0 & 1 & -C_z \end{pmatrix}$$

# Calibration contd..

- Normalizing the homography **T** by making $T_{32} = 1$. Then the camera center in the world coordinate system is

$$C = (C_x, C_y, C_z) = (T_{12}, T_{22}, -T_{11})$$

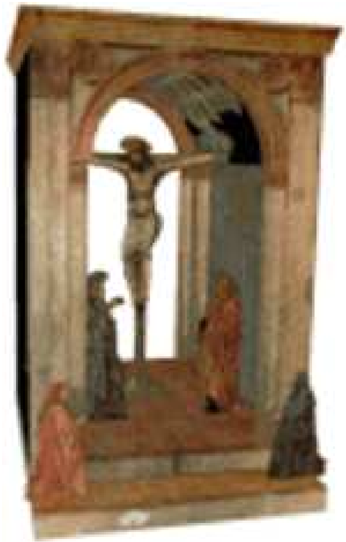
# Computing the 3D coordinates




The height can be computed as

$$h = \text{Camera height} \ast \frac{FH}{CH}$$

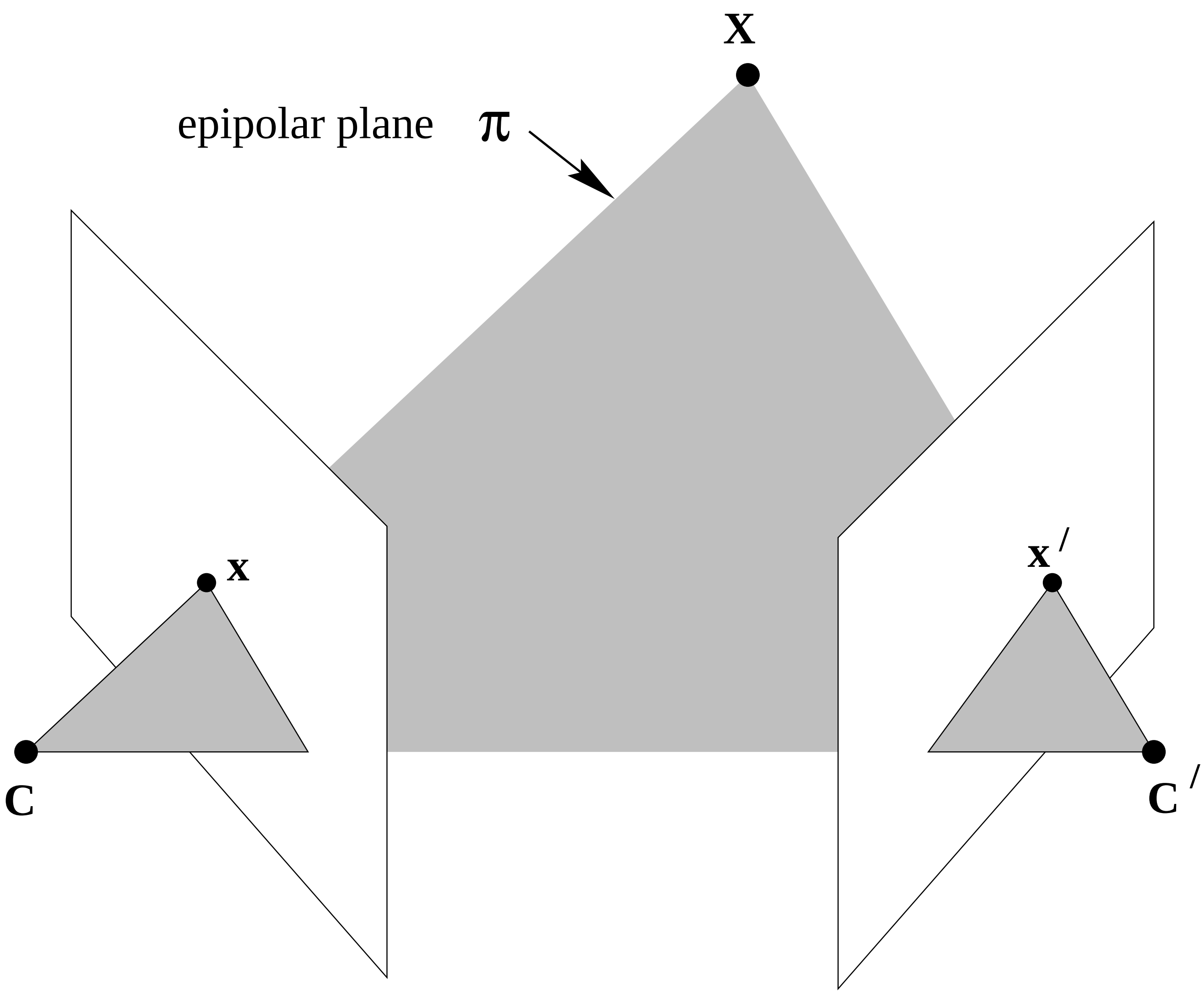
# Results



# Methods of depth estimation

- Stereo: dense correspondence
- Structure from motion
- Depth from focus: needs more photographs
- Depth from defocus: needs careful settings of the camera
- Depth from multiple views

# Point correspondence geometry



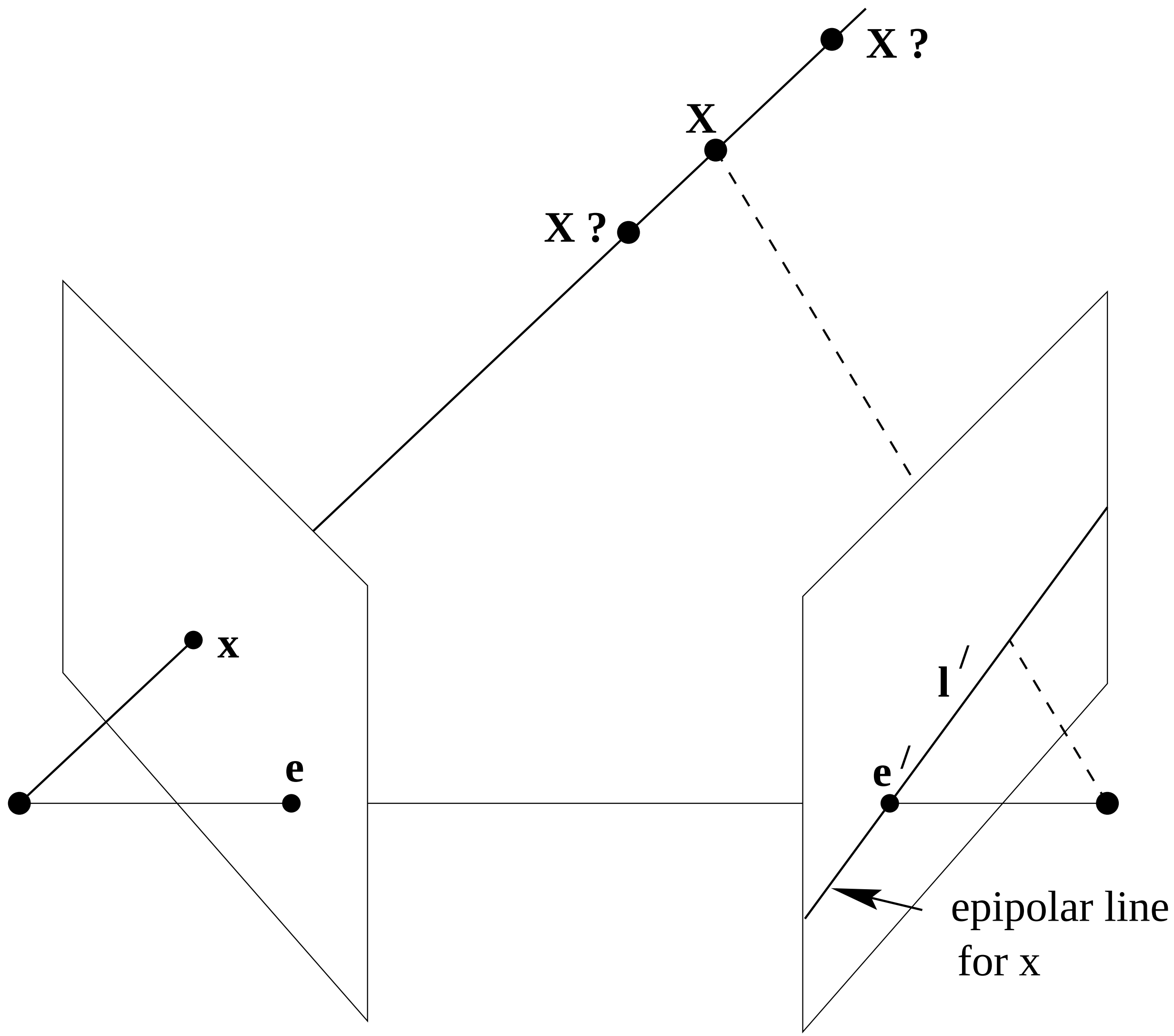
# Epipolar geometry

- $\mathbf{X}$ in space is imaged in two views at $x$ and $x'$
- Aim of stereo: correspondence between $x$ and $x'$
- Baseline: line joining the camera centres
- Intersection of image planes with the pencil of planes having baseline as axis

# Epipolar geometry contd..

- Points $x$, $x'$, $\mathbf{X}$ and camera centers are coplanar, lying in $\pi$
- The back projected rays from $x$ and $x'$ intersect at $\mathbf{X}$
- Given $x$ how the corresponding point $x'$ is constrained?

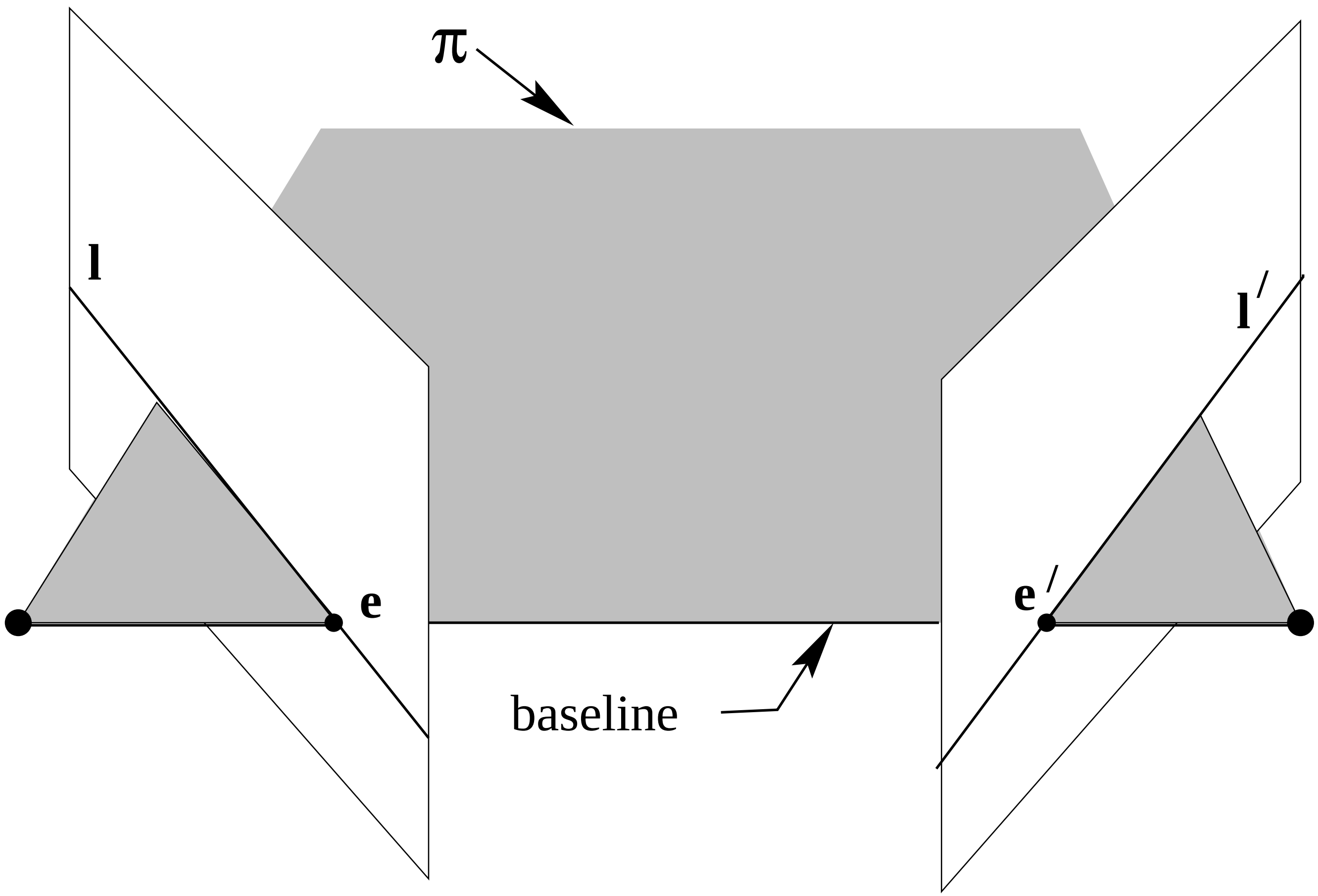
# Point correspondence geometry cont..



# Epipolar geometry contd..

- An image point $x$ back projects to a ray in 3 space defined by the first camera center $C$ and $x$
- This ray is imaged as a line $l'$ in the second view
- The 3-space point $\mathbf{X}$ which projects to $x$ must lie on this ray, so the image of $\mathbf{X}$ in the second image must lie on $l'$

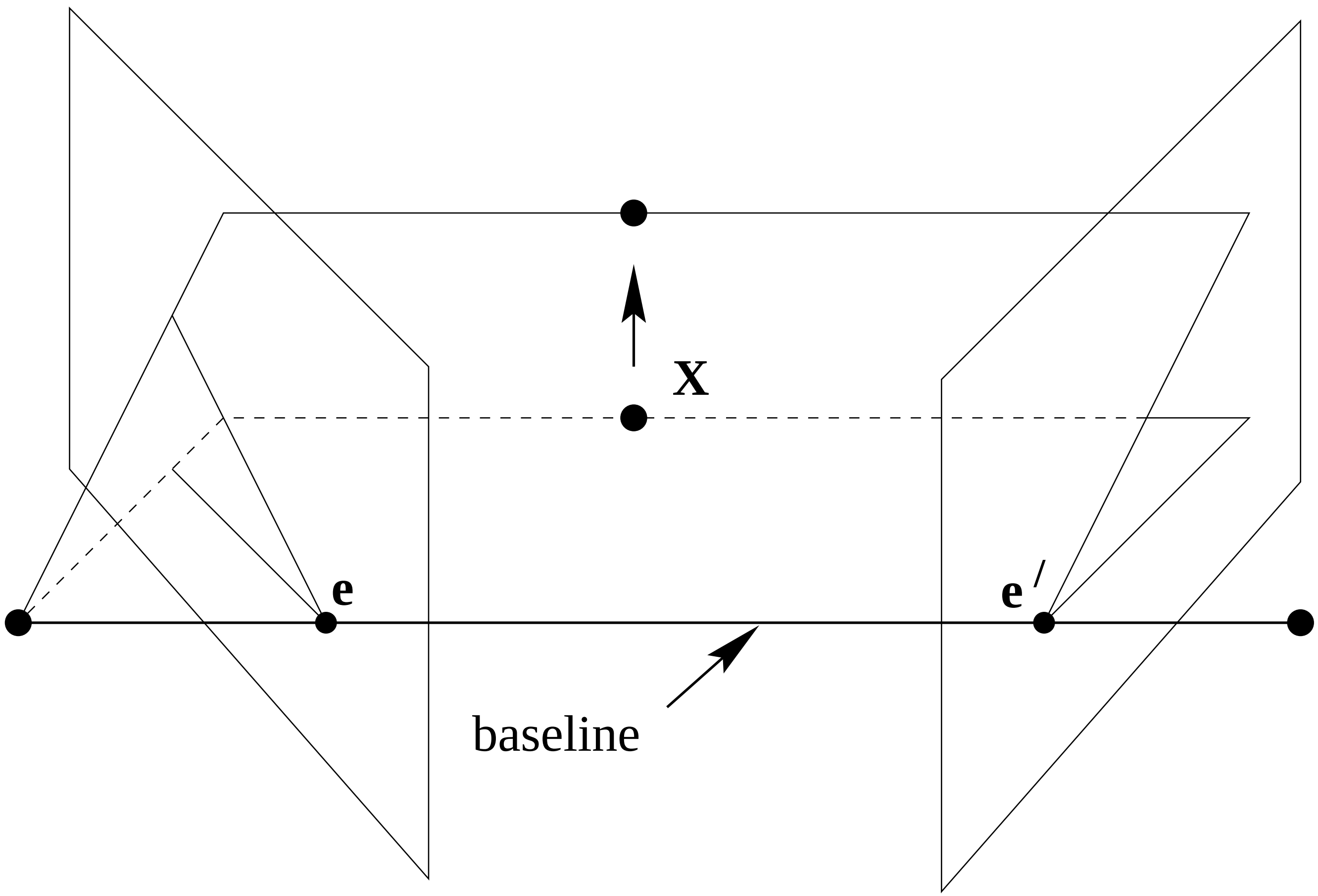
# Epipolar geometry contd..



# Epipolar geometry contd..

- The ray corresponding to the $x'$ lie in $\pi$, hence point $x'$ lies on line of intersection of $l'$ of $\pi$ with the second image plane
- This line $l'$ is the image in the second view of the ray back projected from $x$
- The correspondence is now restricted to the line $l'$

# Epipolar geometry cont..



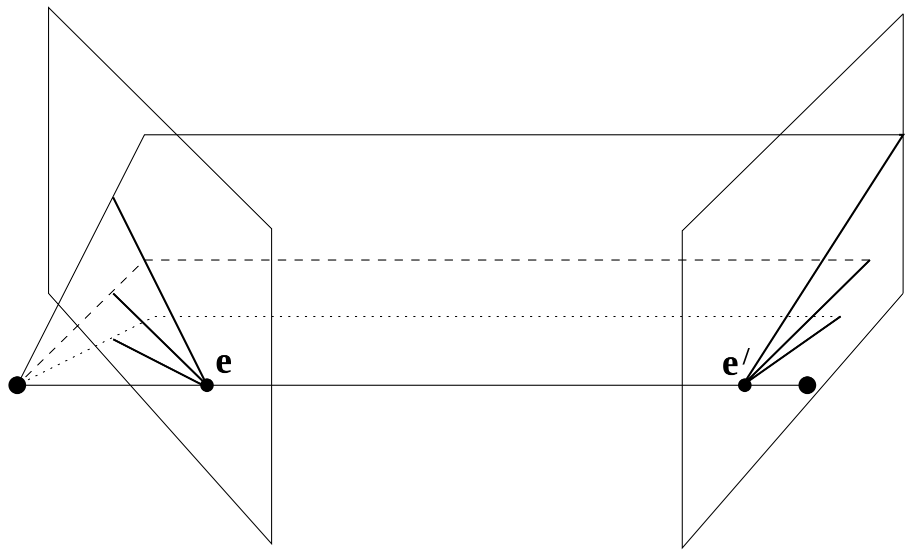
# Epipolar geometry contd..

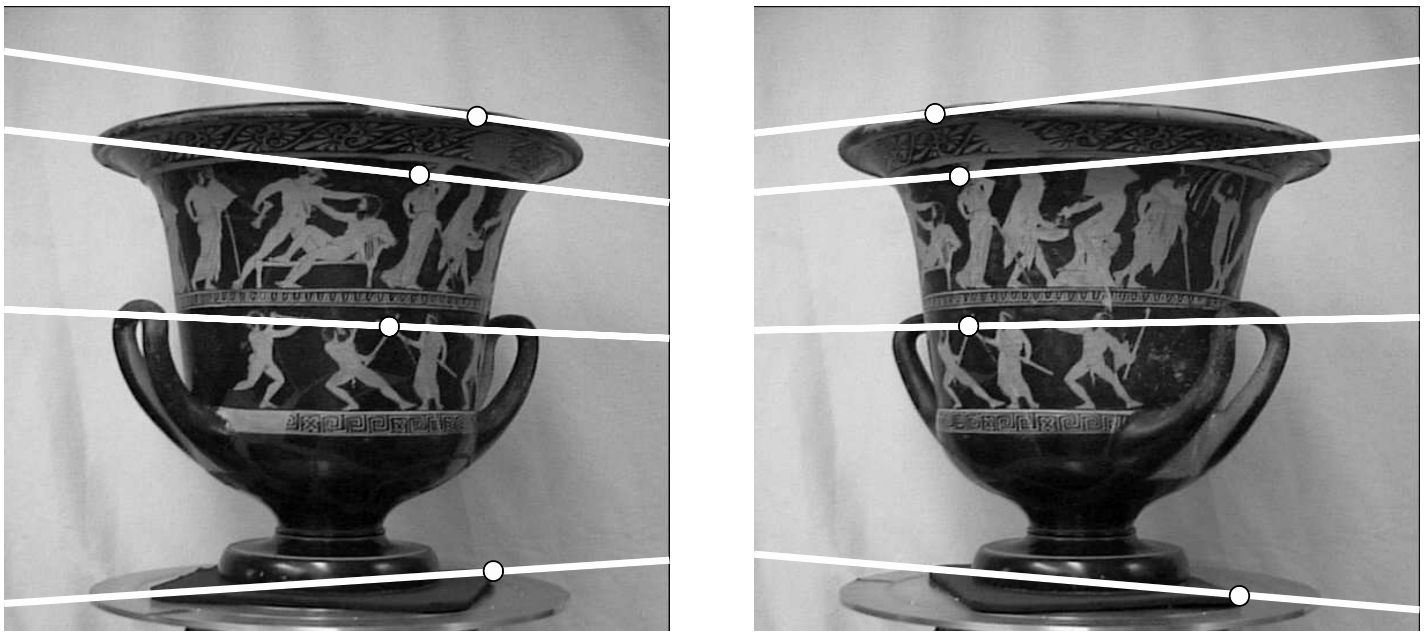
- As the position of the 3D point varies, the epipolar planes rotate about the baseline
- This family of planes is known as an epipolar pencil
- All the epipolar lines intersect at the epipole

# Geometric entities

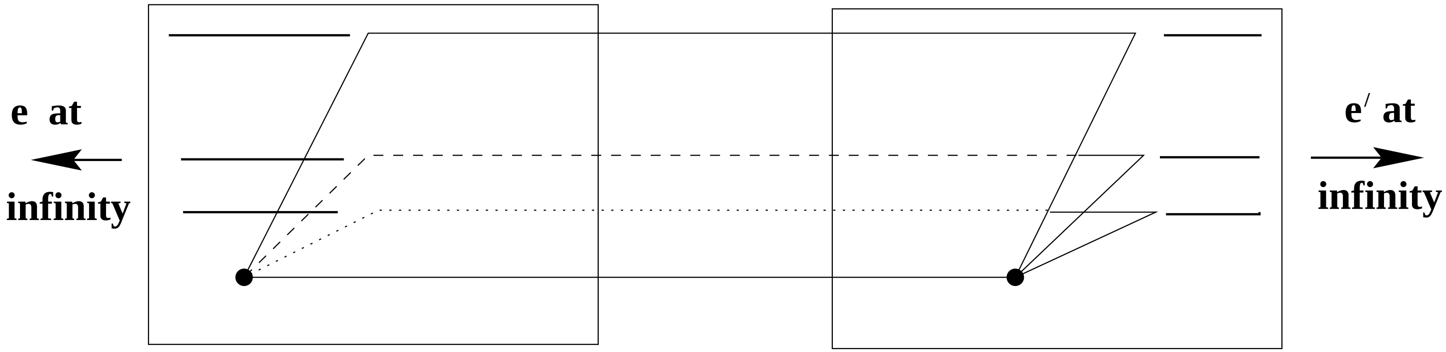
- Epipole: point of intersection of the line joining the camera centres with the image plane
- Epipolar plane: plane containing the baseline
- An Epipolar line: intersection of an epipolar plane with the image plane

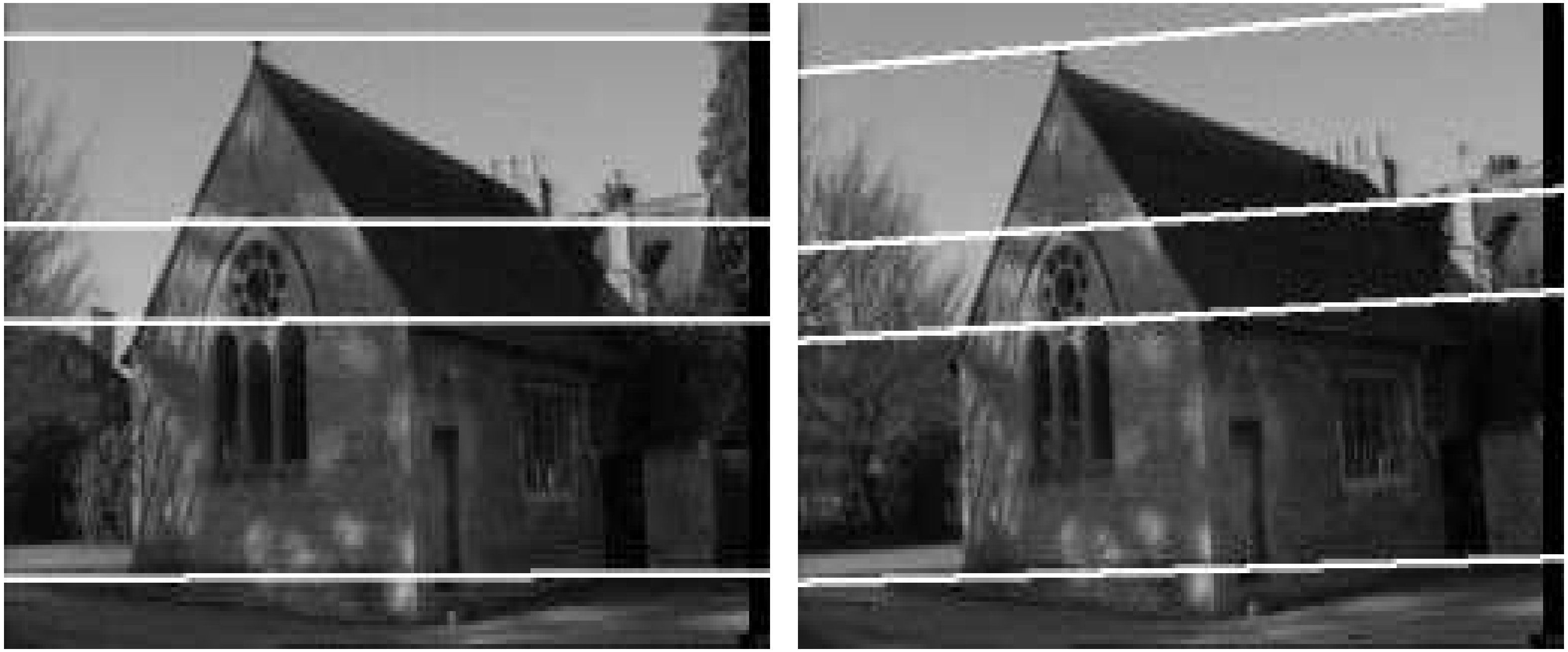
# Converging cameras



# Motion parallel to the image plane



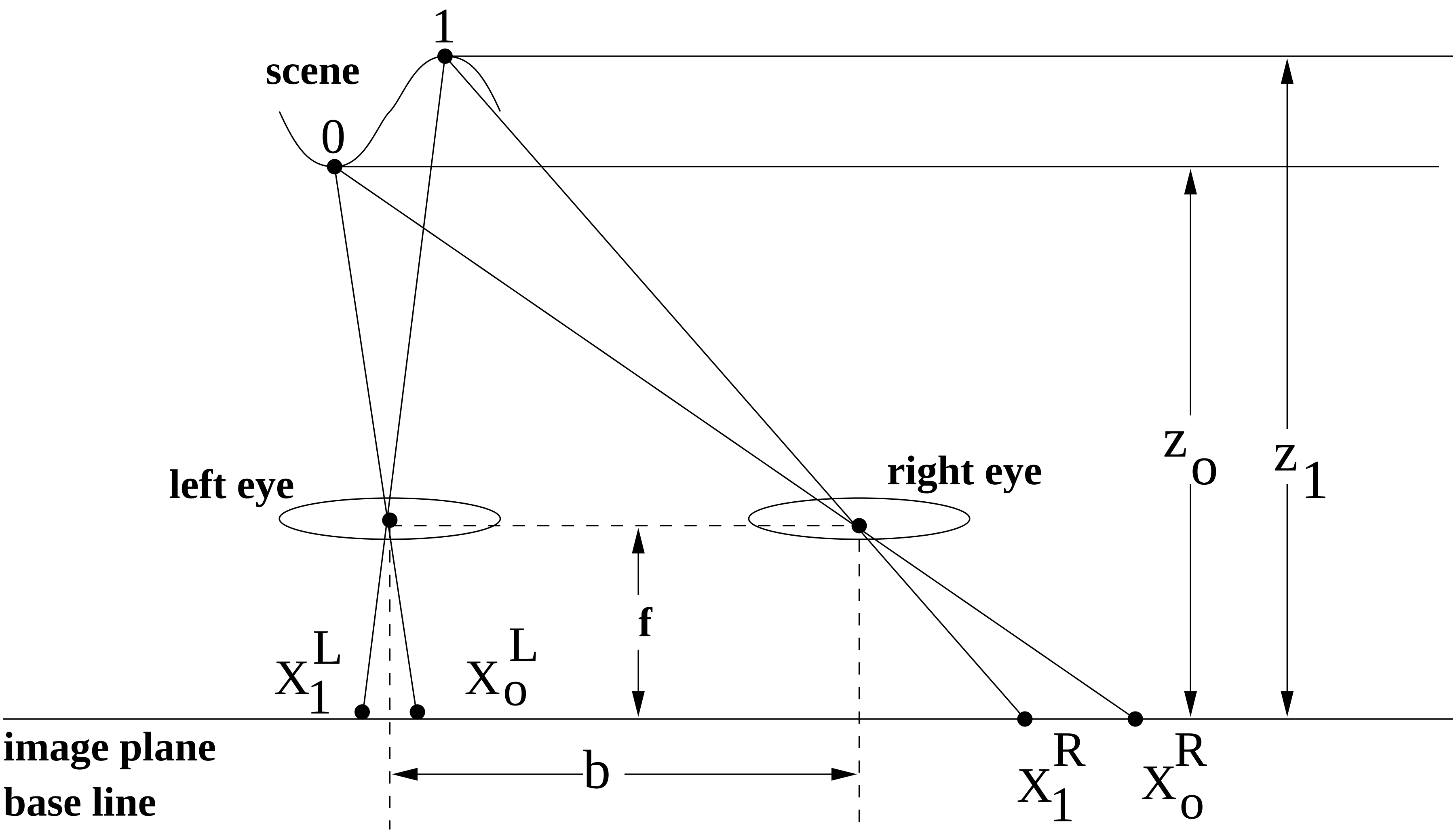
# The Fundamental matrix F

- Algebraic representation of Epipolar geometry.
- For each point **x** in an image, a corresponding epipolar line **l** exists in the other image
- The fundamental matrix satisfies the condition that for any pair of corresponding points $x \leftrightarrow x'$ in the two images $x'^T F x = 0$

# ebraic representation of Fundamental ma

- The Fundamental matrix is given as $\mathbf{F} = \mathbf{K}'^{-\mathbf{T}}\mathbf{R}\mathbf{K}^{\mathbf{T}}[\mathbf{e}]$, where $diag(f, f, 1)$ is a diagonal matrix and $[I|0]$ represents a matrix divided up to $3 \times 3$ block (identity) plus a column vector
- $F$ can be computed in-terms camera matrices for each view

# Stereo



1
scene
0
left eye
right eye
$z_o$
$z_1$
f
$X_1^L$
$X_o^L$
image plane
base line
b
$X_1^R$
$X_o^R$

# Stereo contd..

- We can get

$$Z = \frac{bf}{x^R - x^L}$$

- $b$-baseline, $f$-focal length
  $x^R - x^L$- disparity

- Disparity is inversely proportional to $Z$

# Stereo contd..

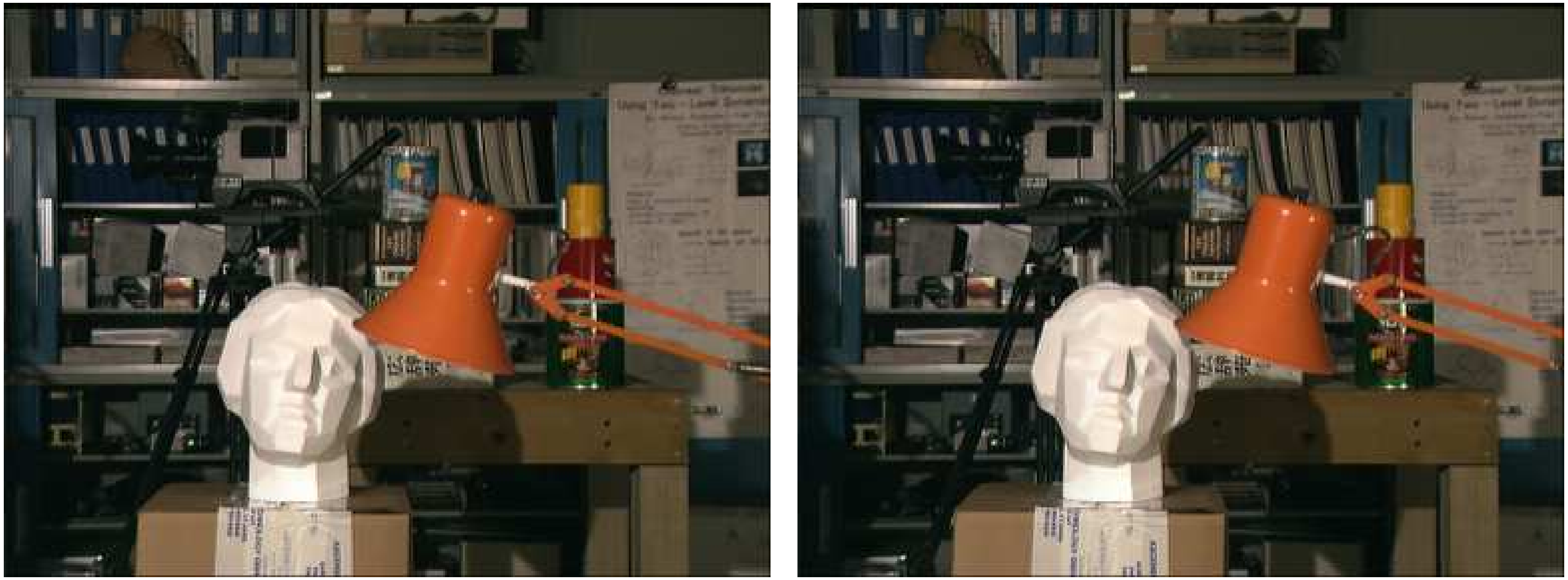
- In intensity based disparity estimation, the disparity $d_{ij}$ is computed by by minimizing

$$min \quad \sum (x^R_{i,j} - x^L_{i,j+d_{ij}})^2$$

where $(x^R_{ij})$ and $(x^L_{ij})$ are the intensity at the pixel location (i,j) in the right image and the left image respectively

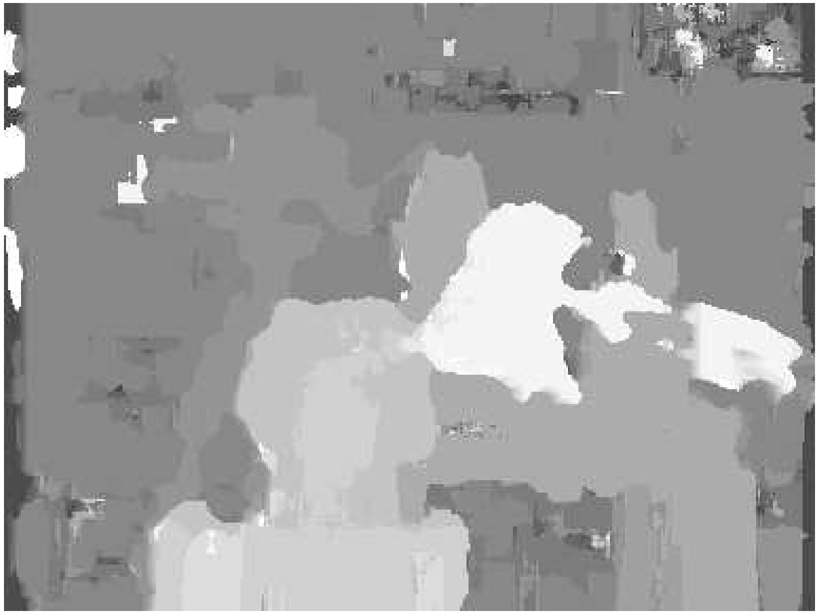
# Results

## Right and left input images



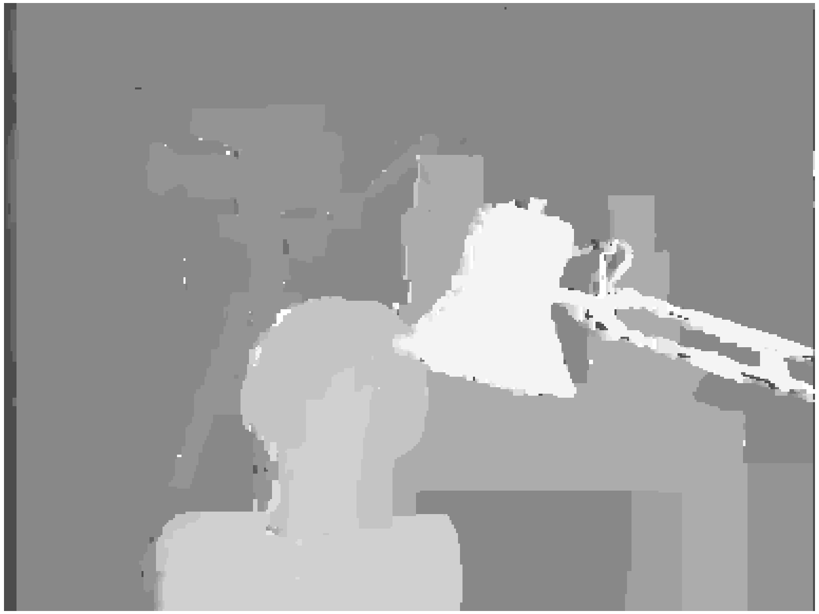
# Results contd..

Depth using correlation method



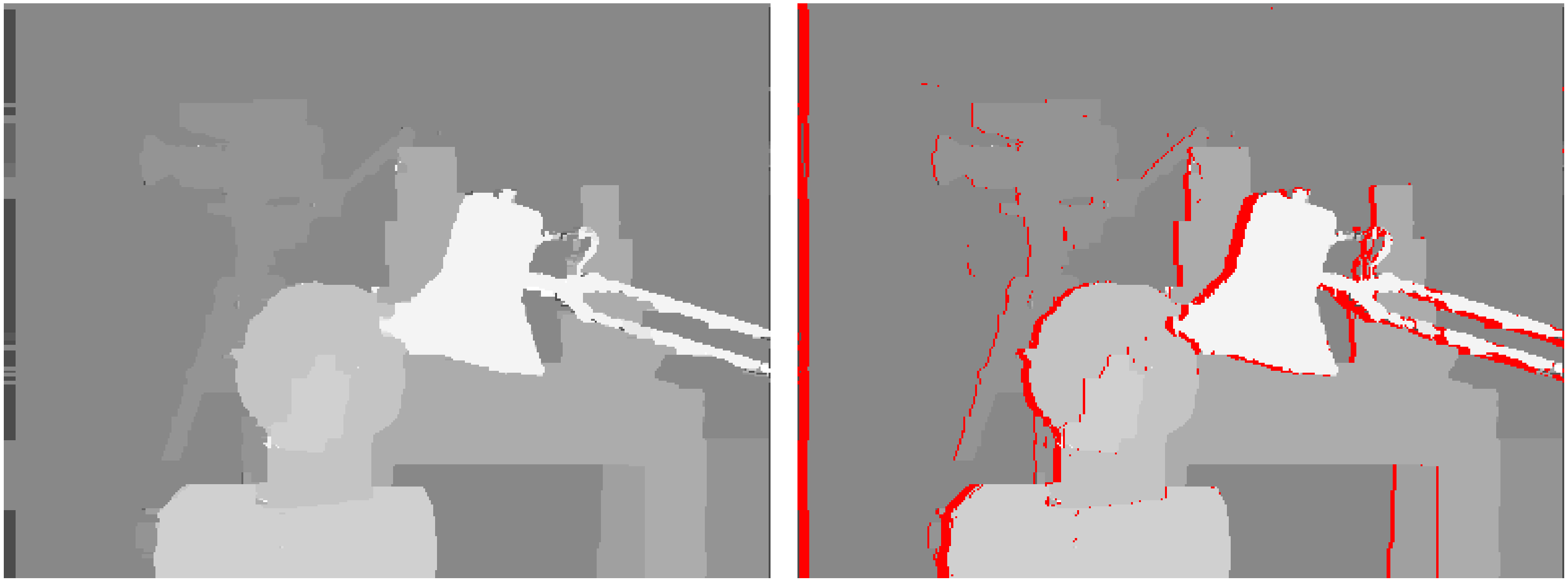
# Results contd..

Depth using BVZ method



# Results

Depth using KZ2 method



# Outline

- Multiview 3D reconstruction
- Problem formulation
- Energy function
- Graph construction
- Algorithm
- Results
- Conclusions

# Multiview 3D reconstruction

- Classic vision problem
- Multiple images of the same scene are used.
- Harder due to reasoning about visibility.
- Very few scene elements are visible from every camera. So visibility cannot be ignored.

# Problem formulation

- Given are $n$ calibrated images of the same scene taken from different viewpoints.
- Let $P_i$ be the set of pixels in the camera $i$, and let $P = P_1 \cup \ldots \cup P_n$ be the set of all pixels.
- A pixel $p \in P$ corresponds to a ray in 3D-space.
- Consider the point of the first intersection of this ray with an object in the scene.
- Goal is to find the depth of this point for all pixels in all images.
- A pair $\langle p, l \rangle$ where $p \in P$, $l \in L$ corresponds 3D-point.

# Energy function

- The energy function consists of three terms

$$E(f) = E_{data}(f) + E_{smoothness}(f) + E_{visibility}(f)$$

- The data term will impose photo-consistency.

$$E_{data}(f) = \sum_{\langle p, f(p)\rangle, \langle q, f(q)\rangle \in I} D(p,q)$$

where

$$D(p,q) = min\{0, (Intensity(p) - Intensity(q))^2 - K\}$$

$I$ is the set of 3D-points satisfying the following:
- Only 3D-points at the same depth can interact, i.e.if $\{\langle p_1, l_1\rangle, \langle p_2, l_2\rangle\} \in I$ then $l_1 = l_2$.

# Smoothness term

- This involves the notion of neighborhood.

$$E_{smoothness}(f) = \sum_{\{p,q\}\in\mathbf{N}} V_{\{p,q\}}(f(p), f(q))$$

where $V_{\{p,q\}}$ is a metric.
$N$ is a 4-neighborhood system.

- This term imposes smoothness while preserving discontinuities.

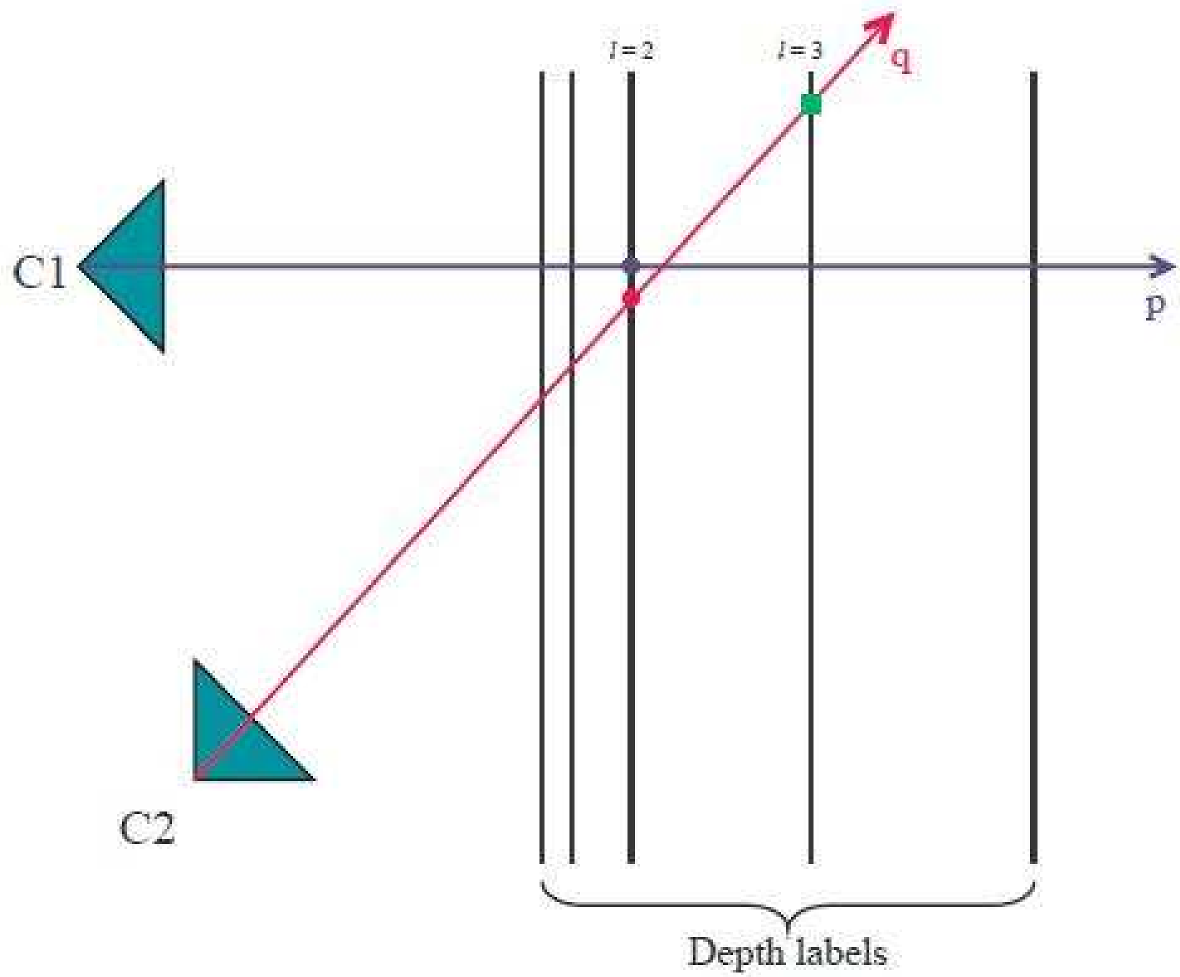
# Visibility term

- This term will encode the visibility constraint. It is zero if the constraint is satisfied; otherwise infinity.

$$E_{visiblity}(f) = \sum_{\langle p, f(p) \rangle, \langle q, f(q) \rangle \in I_{vis}} \infty$$

  where $I_{vis}$ satisfies:
  - Only 3D-points at different depths can interact, i.e. if $\langle p_1, l_1 \rangle, \langle p_2, l_2 \rangle \in I_{vis}$ then $l_1 \neq l_2$.
- The visibility constraint states that the color and intensity of a 3D-point visible in the camera remains same.
- This 3D point may block views from other cameras. If a 3D-point $\langle p, l \rangle$ is present in a configuration $f$, and if a ray corresponding to a pixel $q$ goes through $\langle p, l \rangle$ then its depth is at most $l$.

# Visibility constraint



# Graph construction

- For energy functions of binary variables of the form

$$E(x_1,\ldots,x_n) = \sum_{i<j} E^{i,j}(x_i,x_j)$$

  it is possible to construct a graph for minimizing it if and only if each term $E^{i,j}$ satisfies the following condition:

$$E^{i,j}(0,0) + E^{i,j}(0,0) \leq E^{i,j}(0,1) + E^{i,j}(1,0).$$

- Then the graph $G$ is constructed as follows:
  - Add a node $v_i$ for each variable $x_i$.

# Adding edges to the graph

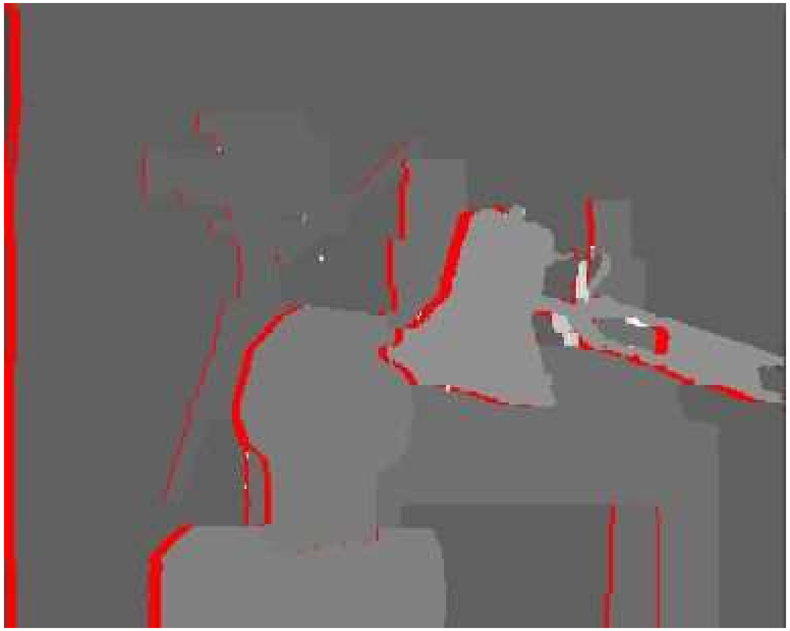
- For each term $E^{i,j}(x_i, y_j)$, add the edges as follows:
  - If $E(1,0) > E(0,0)$ then add an edge $(s, v_i)$ with the weight $E(1,0) - E(0,0)$, otherwise add an edge $(v_i, t)$ with the weight $E(0,0) - E(1,0)$.
  - if $E(1,0) > E(1,1)$ then add an edge $(v_j, t)$ with the weight $E(1,0) - E(1,1)$, otherwise add an edge $(s, v_j)$ with the weight $E(1,1) - E(1,0)$.
  - The last edge that is added is $(v_i, v_j)$ with the weight $E(0,1) + E(1,0) - E(0,0) - E(1,1)$.

# Algorithm

- The algorithm is straightforward.
  - Select(in a fixed order or at random) a disparity α.
  - Find a unique configuration within a single α-expansion move(local improvement step).
  - If this decreases the energy,then go to that label; If there is no α that decreases the energy then we are done.
  - One restriction on the algorithm is that the initial configuration must satisfy the visibility constraint
  - The critical step is to efficiently compute the α-expansion with the smallest energy. This is accomplished using graph cuts by solving minimum cut problem.

# Results: Data set 1

# Results for data set 1



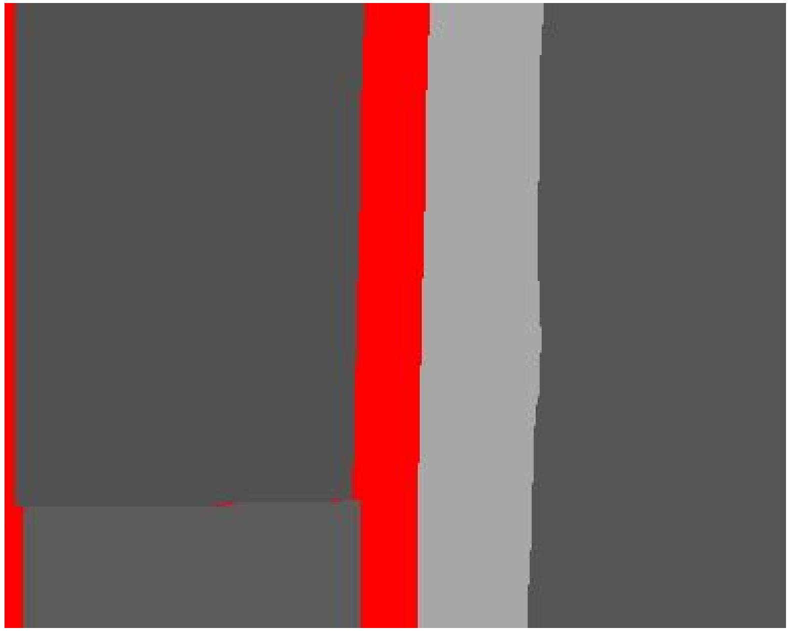
# Data set 2




# Results for data set 2

# Data set 3

# Results for data set 3

# Conclusions

- Graph cuts used for optimizing the energy function.
- Gives good reconstruction for less occlusions.