Stereo

Jenn-Jier James Lien (連 震 杰)
Professor

Computer Science and Information Engineering
National Cheng Kung University

(O) (06) 2757575 ext. 62540
jjlien@csie.ncku.edu.tw
http://robotics.csie.ncku.edu.tw

Major Issues

1. **Stereo system: Find depth**

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

   1) **Camera calibration:**
      (1) Intrinsic parameters
      (2) Extrinsic parameters
      (3) Lens distortion parameters
   2) **Rectification**
   3) **Epipolar geometry and Epipolar constraint:**
      - Reduce 2D search to 1D search
   4) **Essential matrix and Fundamental matrix:** To find corresponding point

2. **Corresponding point: Matching – Similarity measure**
   1) Window size
   2) Search range
   3) Match metrics

3. **Match techniques: :**
   1) Dynamic Programming
   2) Graphcuts
   3) …

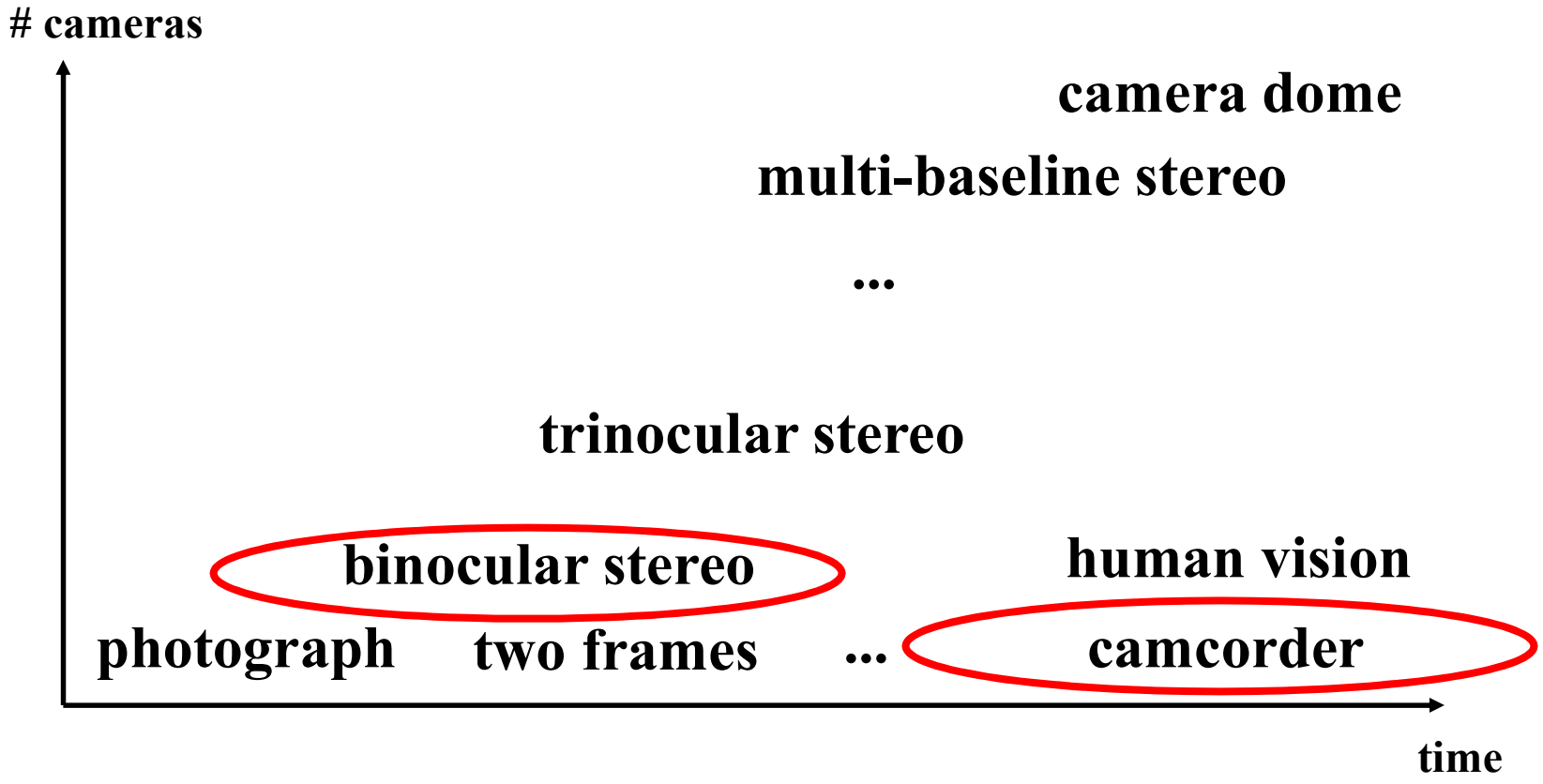
3. **Disparity map: Recover depth from disparity map**

Stereo Procedure

1. **Camera calibration:**
   - **1) Intrinsic parameters**
   - **2) Extrinsic parameters**
   - **3) Lens distortion parameters**
2. **Rectification:**
   - **1) Epipolar geometry and Epipolar constraint:**
     - ➢ Reduce 2D search to 1D search
   - **2) Essential matrix and Fundamental matrix:** To find corresponding point
3. **Find Corresponding point:**
   - **1.1) Matching – Similarity measure**
     - (1) Window size (2) Search range (3) Match metrics
   - **1.2) Match Error (ex. Left-Right Check)**
     - ➢ **Delete match error pixels and then fill the holes**
   - **2) Match techniques: :**
     - (1) Dynamic Programming (2) Graphcuts (3)
4. **Disparity map: Recover depth from disparity map**

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

Modeling from Multiple Views

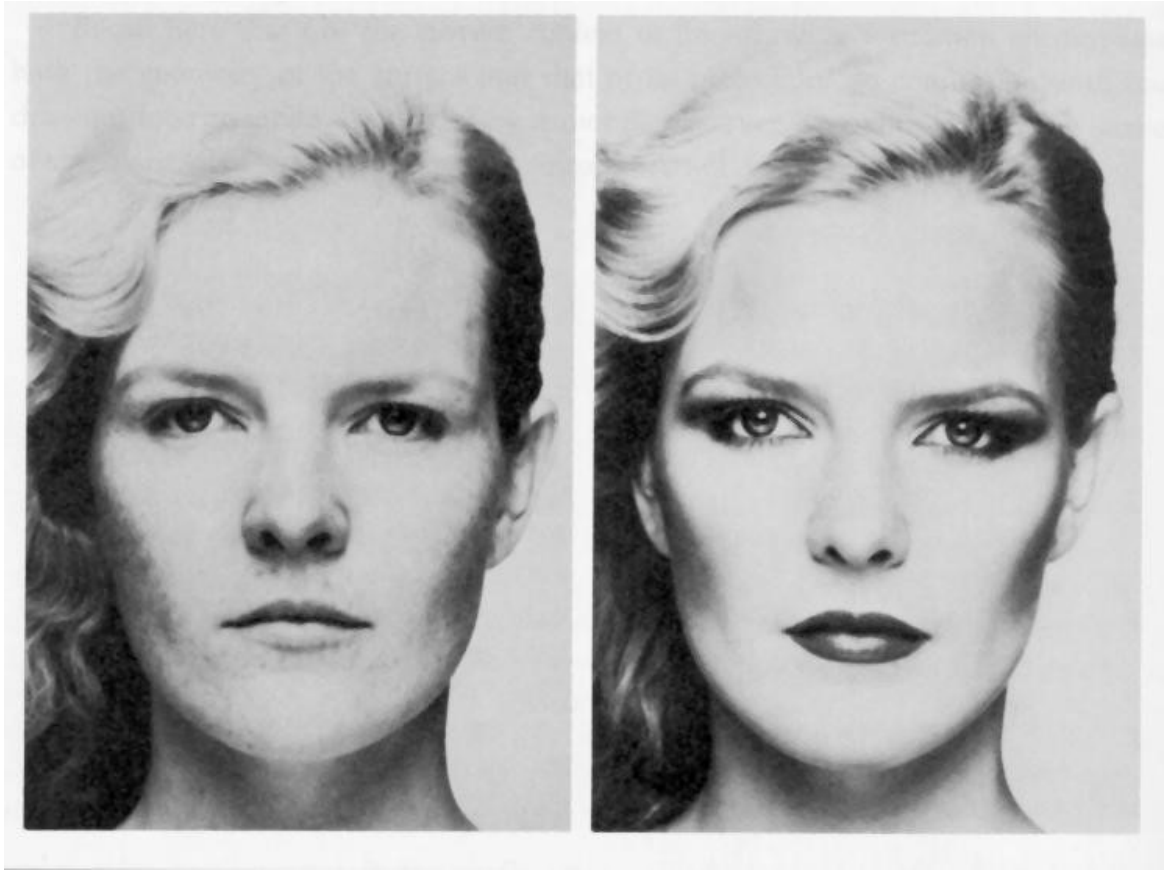


Recovering 3D From Images or Video

- **So far, we've relied on a human to provide depth cues**
  - parallel lines, reference points, etc.
- **How might we do this automatically?**
  - What cues in the image provide 3D information?

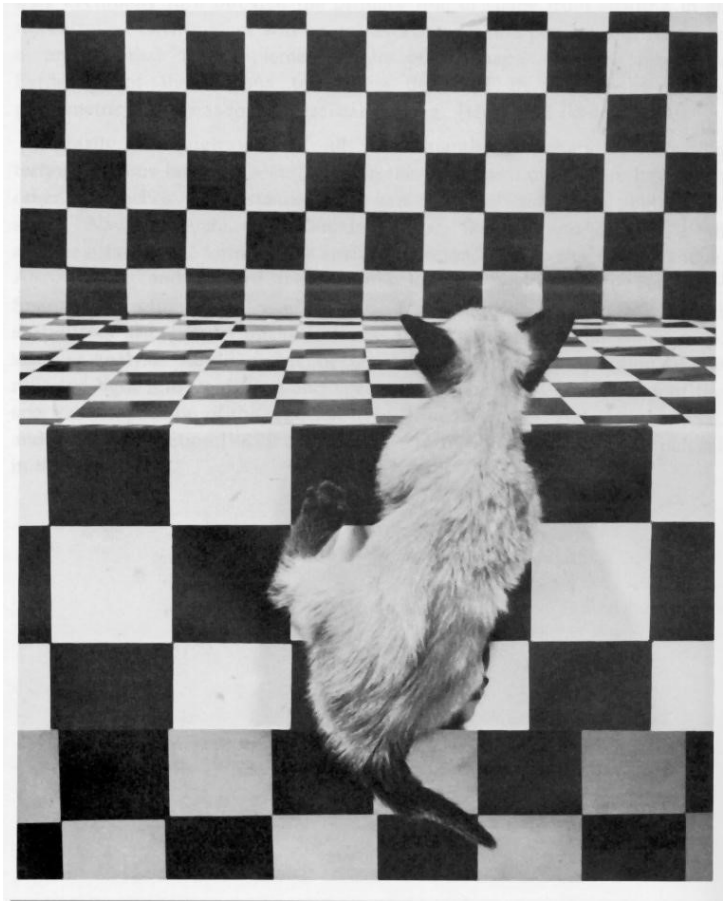
Visual Cues

- **Shading**



Merle Norman Cosmetics, Los Angeles

- Shading
- **Texture**



***The Visual Cliff*, by William Vandivert, 1960**

- Shading
- Texture
- **Focus**

From *The Art of Photography*, Canon

- Shading
- Texture
- Focus
- **Motion**

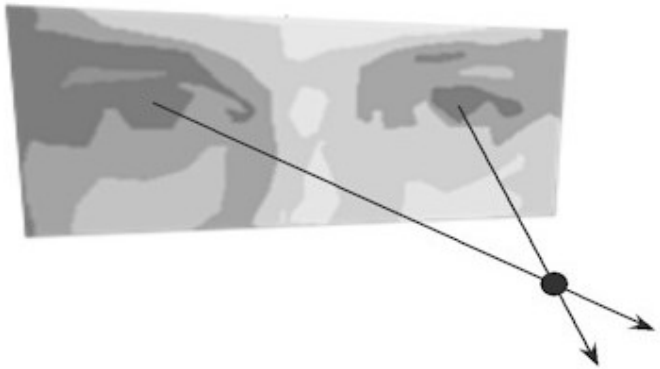
- Shading
- Texture
- Focus
- Motion

- **Others:**
  - Highlights
  - Shadows
  - Silhouettes
  - Inter-reflections
  - Symmetry
  - Light Polarization
  - ...

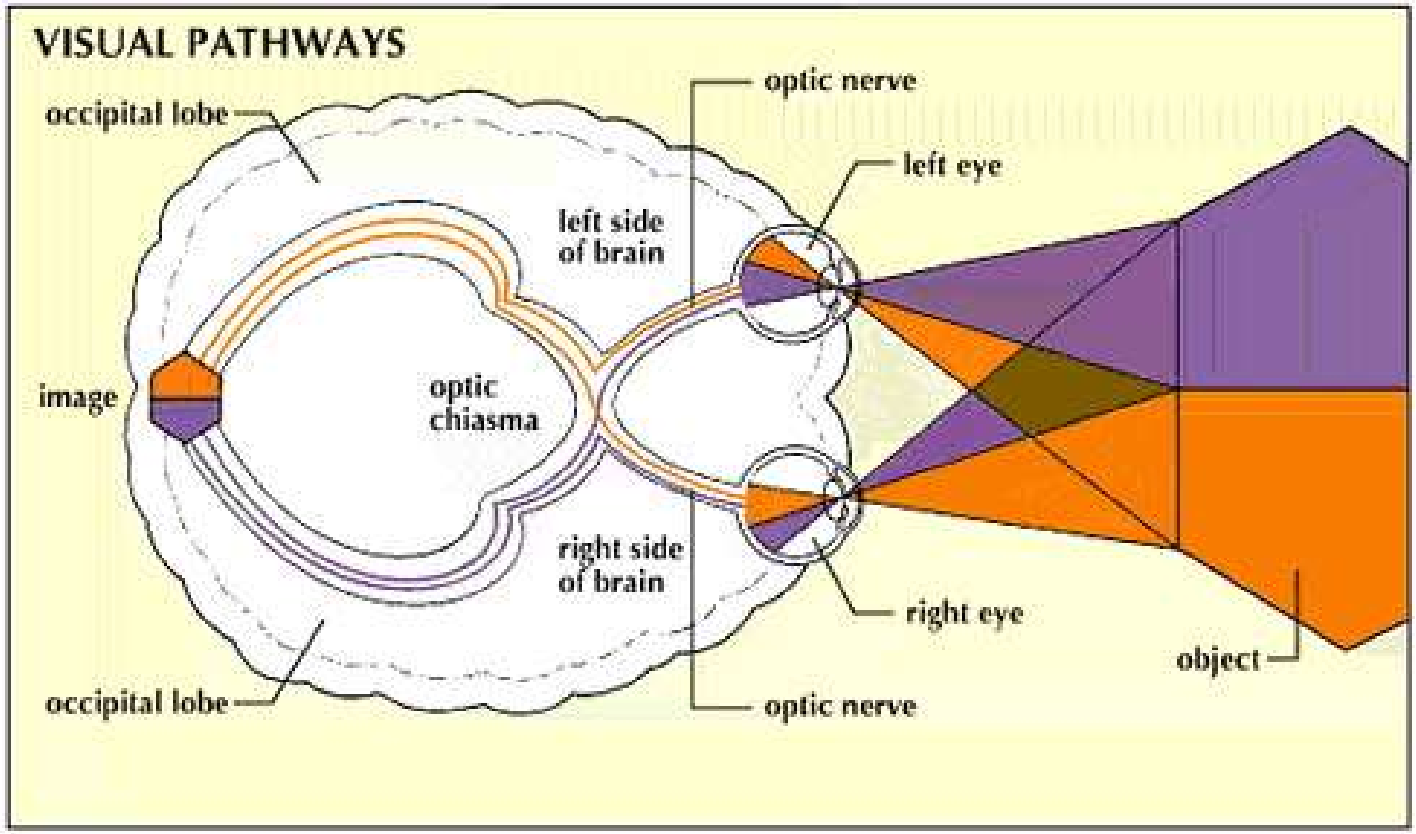
Shape From X

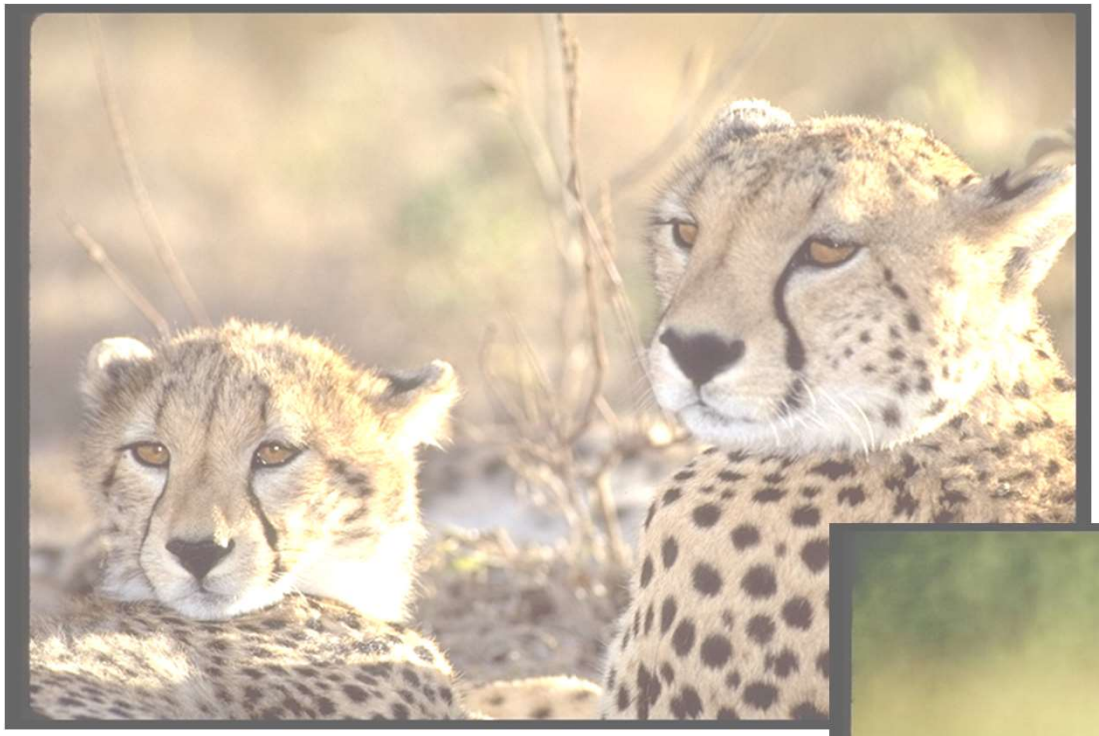
- X = shading, texture, focus, motion, ...
- In this class we'll focus on the motion cue

Stereo Vision (Stereopsis)

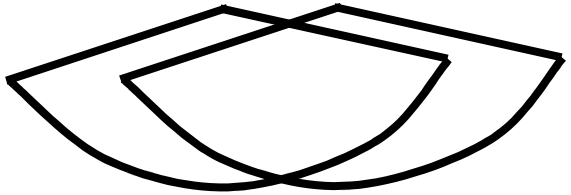


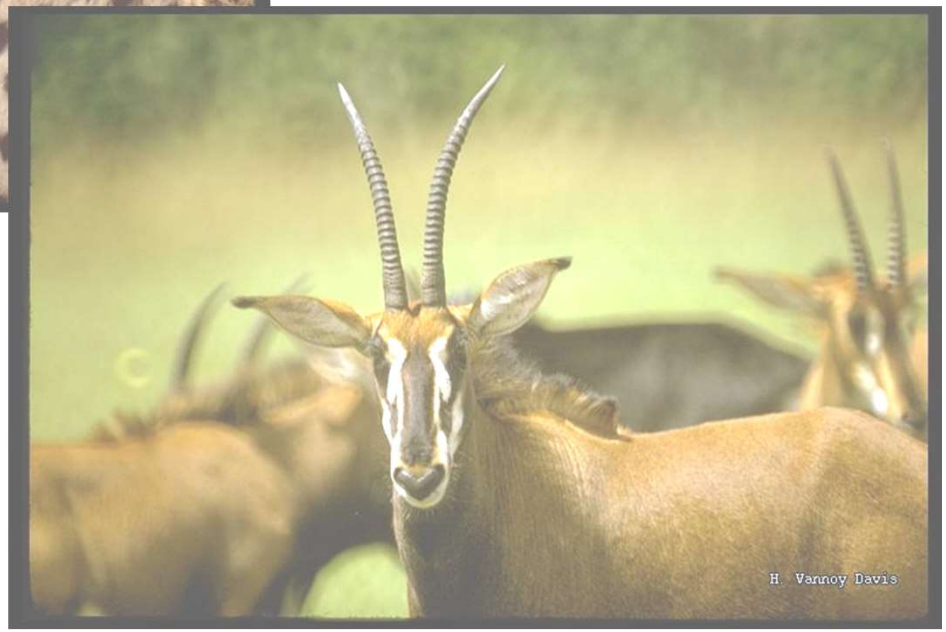
- **Stereo vision is a technique for the reconstruction of the three-dimensional description (or 3D depth) of a scene from images observed from multiple viewpoints.**



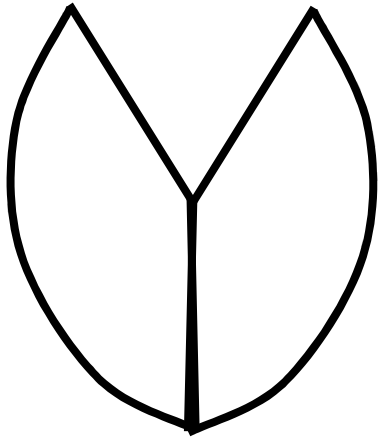


Cheetah

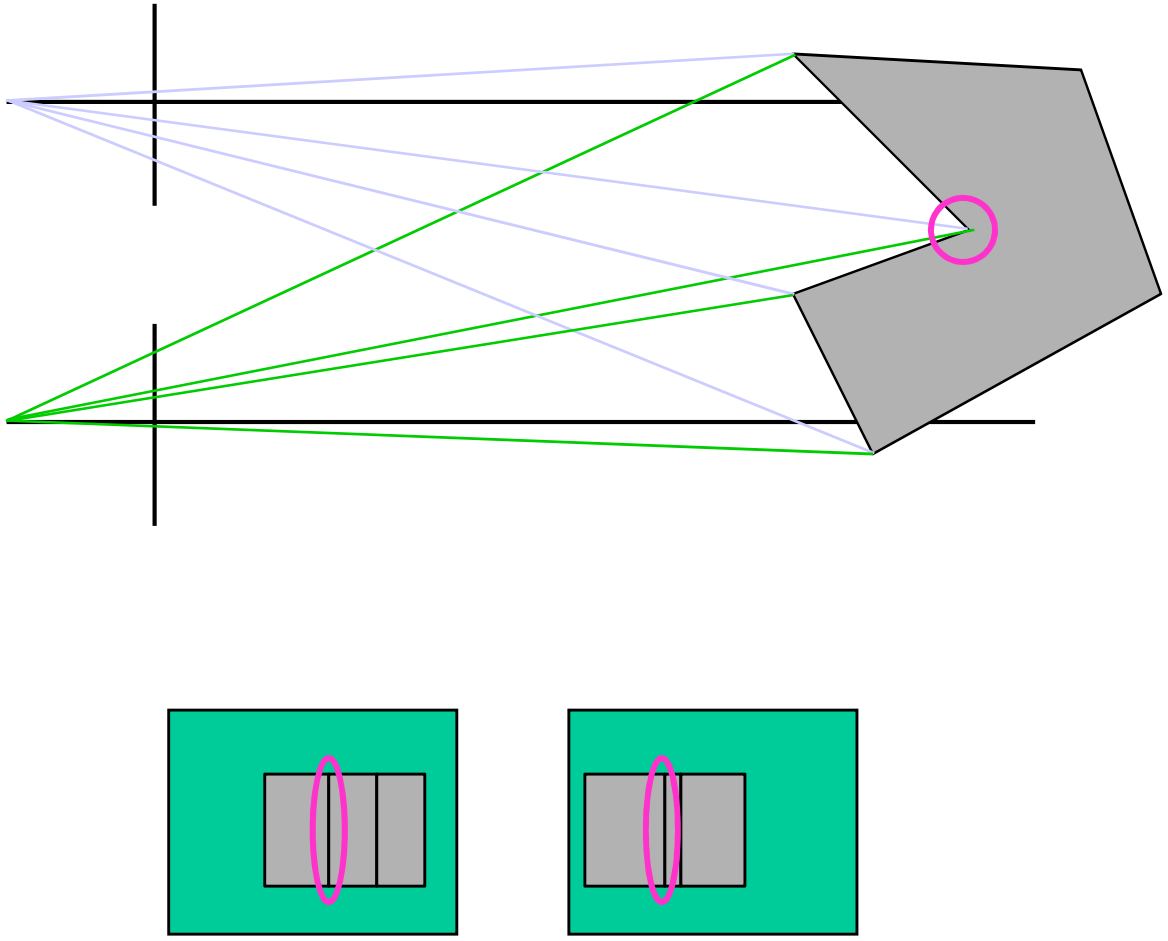




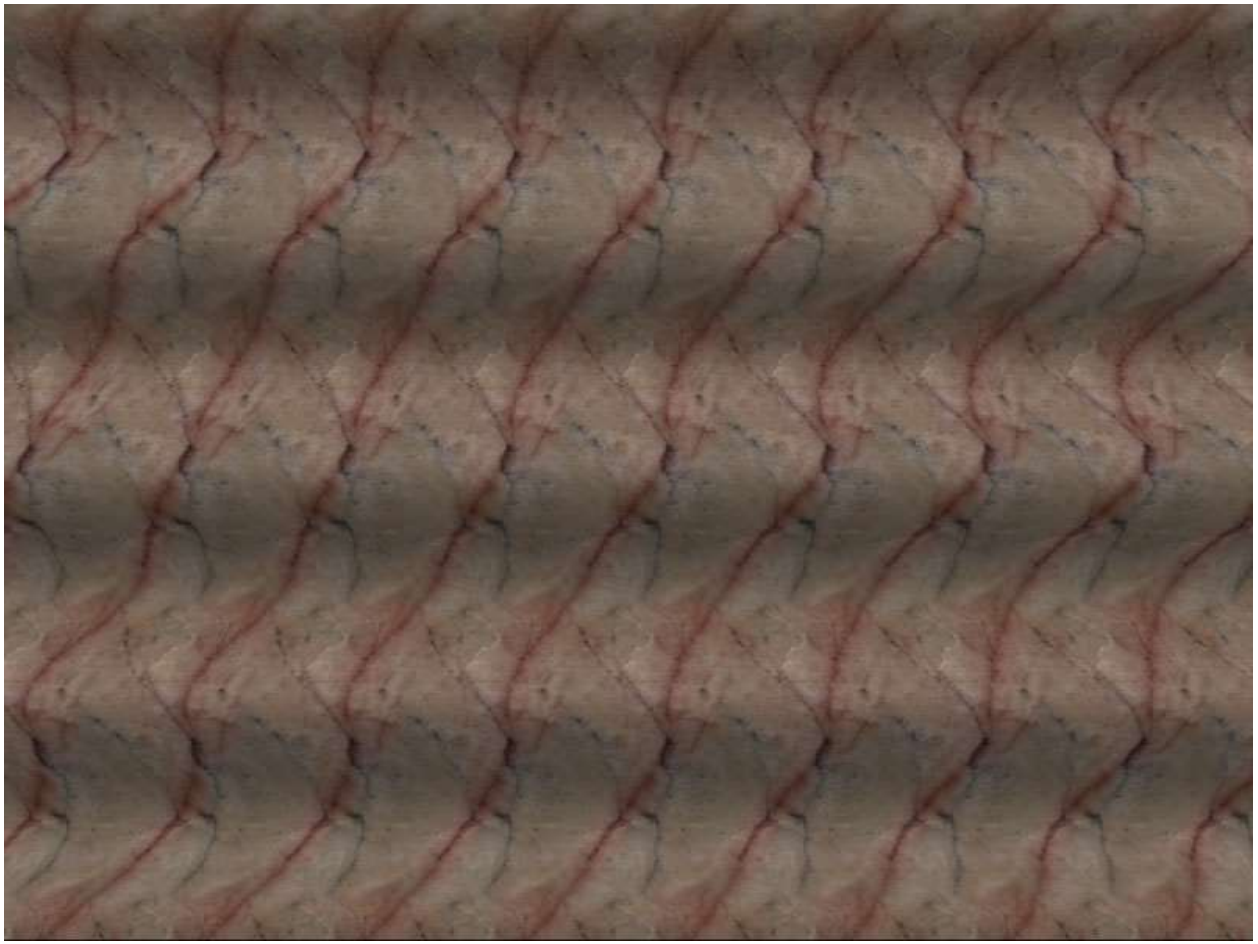
Antelope



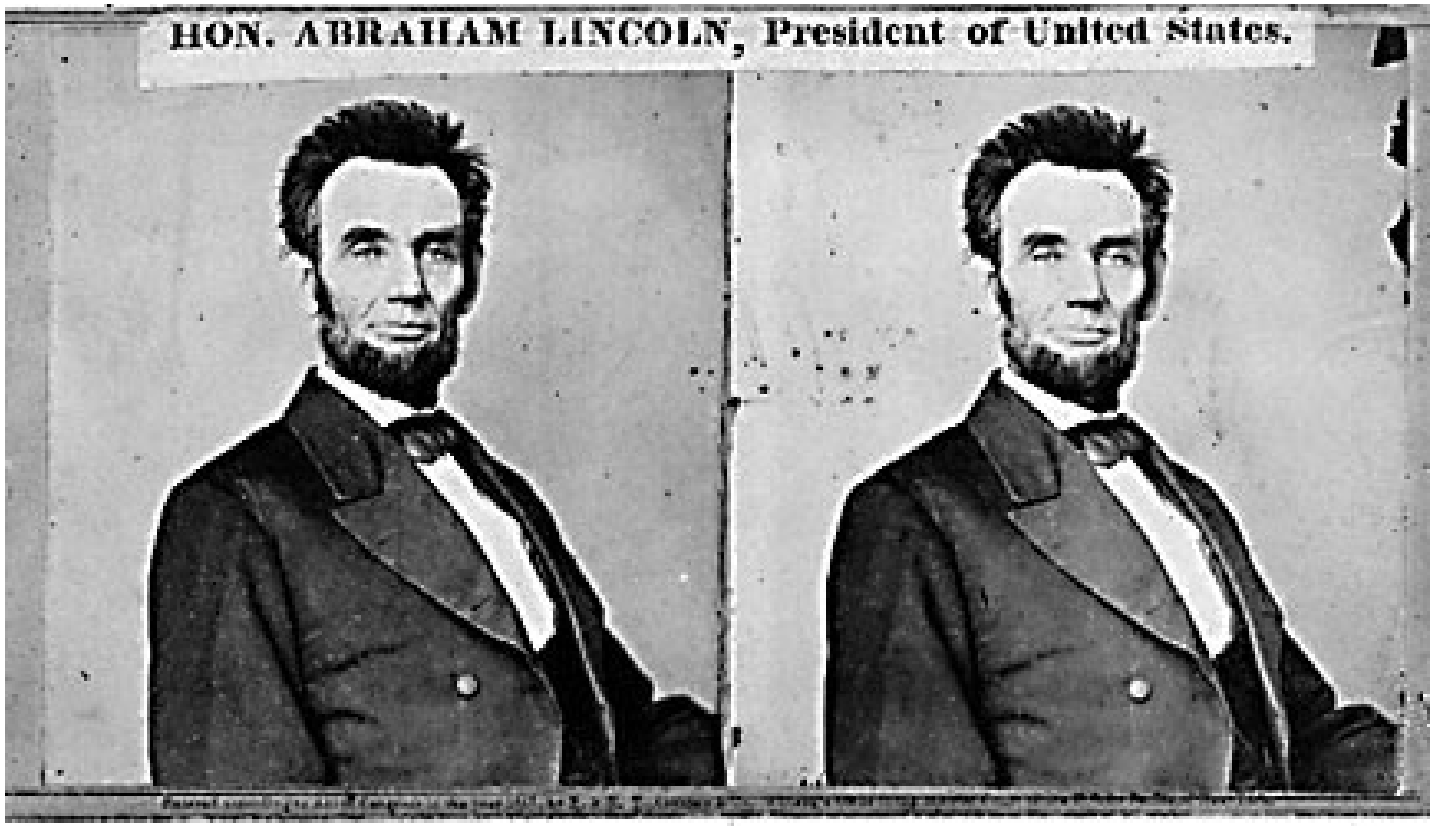
[photos courtesy California Academy of Sciences]



Stereo Photos: Repeated Texture



Single image stereogram, by Niklas Een


HON. ABRAHAM LINCOLN, President of United States.

Public Library, Stereoscopic Looking Room, Chicago, by Phillips, 1923

Teesta suspension bridge-Darjeeling, India

Mark Twain at Pool Table", no date, UCR Museum of Photography

Woman getting eye exam during immigration procedure at Ellis Island, c. 1905 - 1920 , UCR Museum of Phography

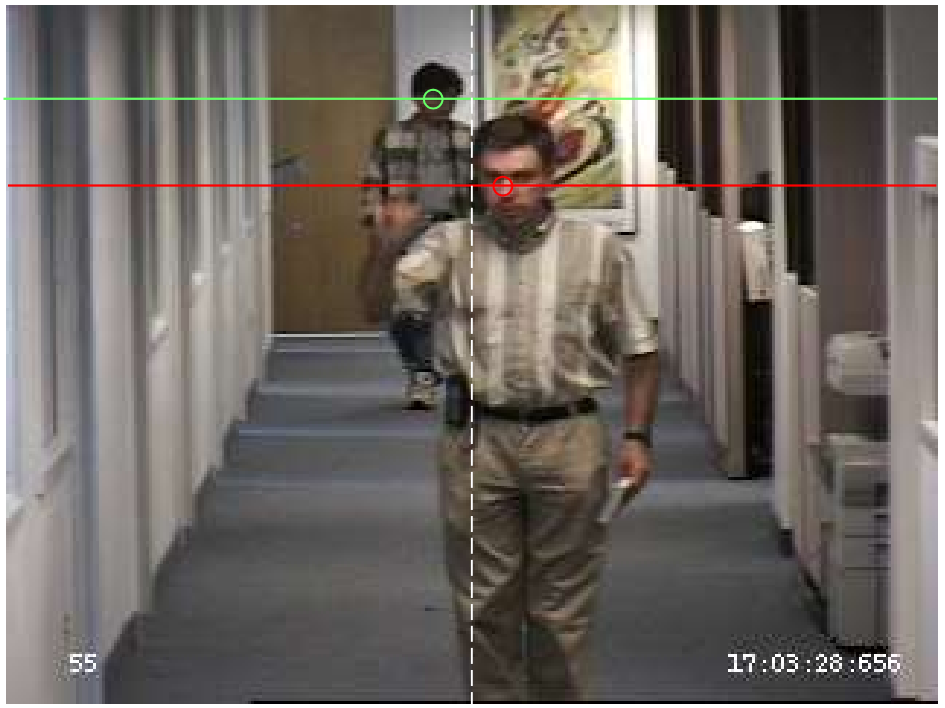
Stereo Basics

- **In a stereo system, we have two images of a scene and the usual goal is to recover**
  - The 3D structure of the scene
  - The relative positions of the two cameras
  - Both of the above

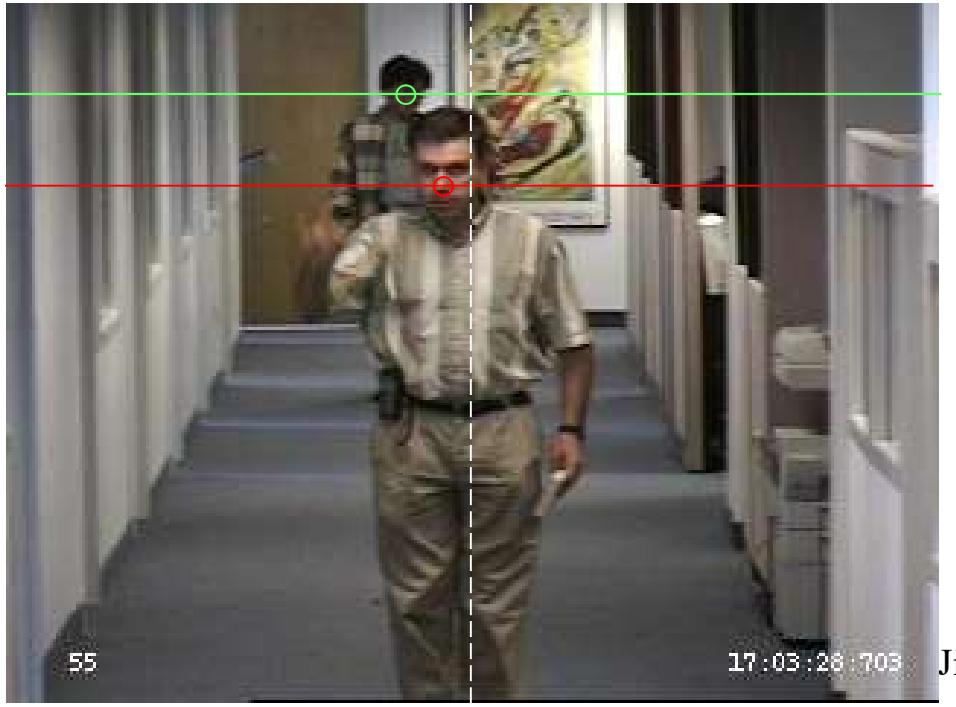
Left Image

Right Image

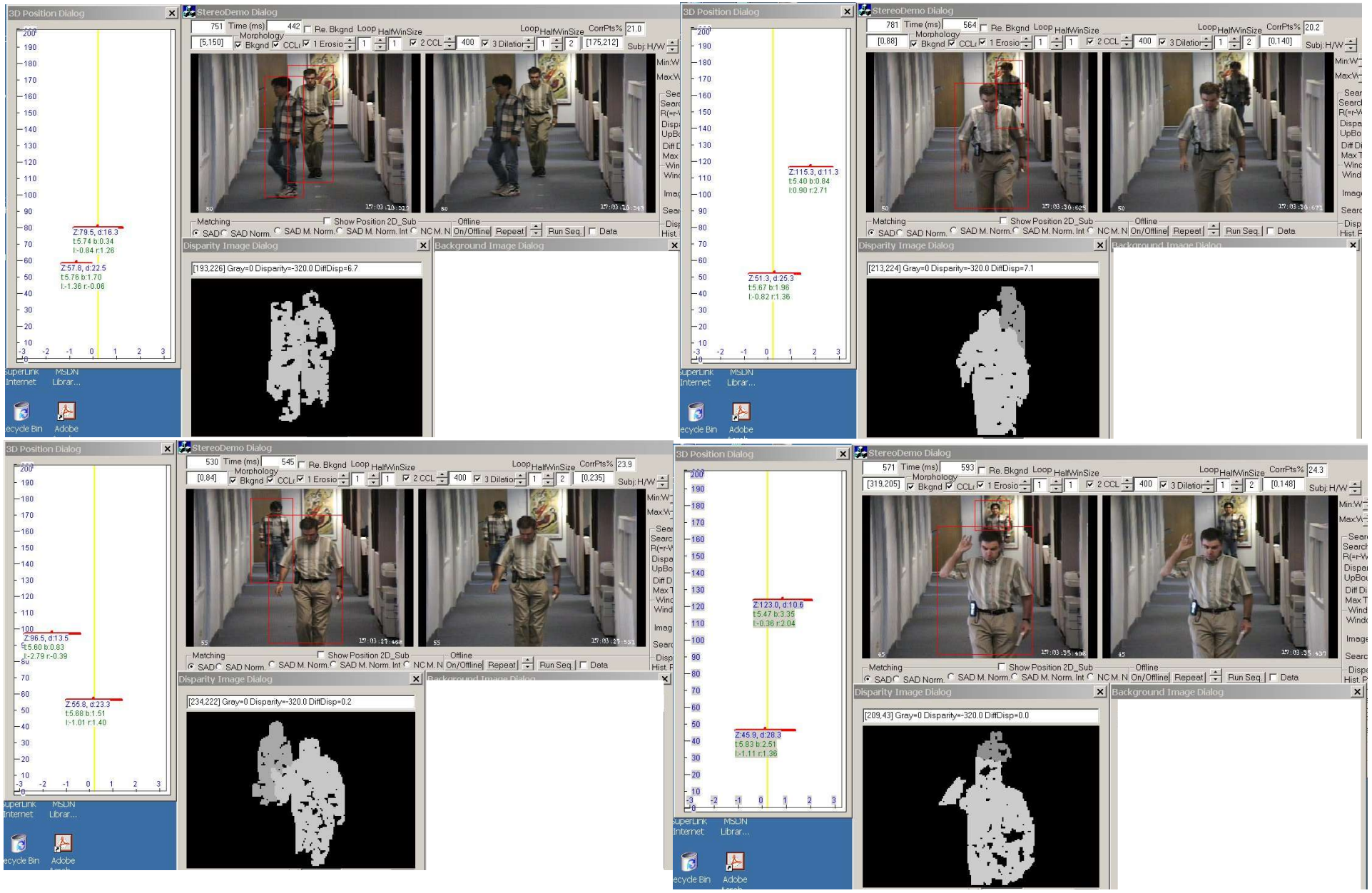
Left Image



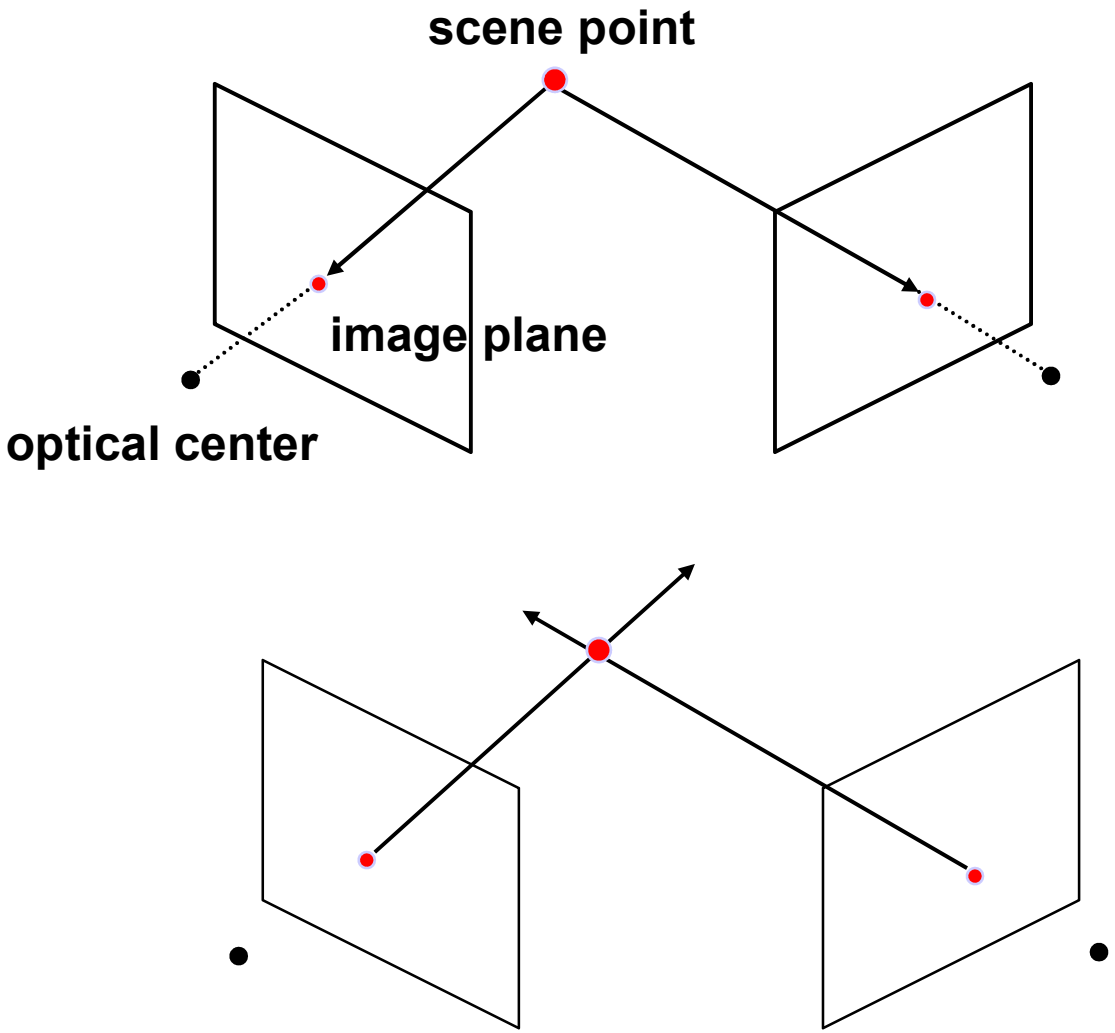
Right Image



Stereo: Body 3D-Position Estimation



Stereo



- **Basic Principle: Triangulation**
  - Gives reconstruction as intersection of two rays
  - Requires
    1) calibration
    2) point correspondence
- **The stereo problem is usually broken into two sub-problems**
  1) The reconstruction problem
     - » Recovering the geometry of the scene and/or the relative camera positions => Panorama
  2) The correspondence problem
     - » which involves identifying corresponding points in both images

Simplest Stereoscopic Imaging Technique I: $Z = \frac{f * B}{d}$

- Stereoscopic =? stereo for short
- Consider two 1D cameras with identical intrinsic parameters

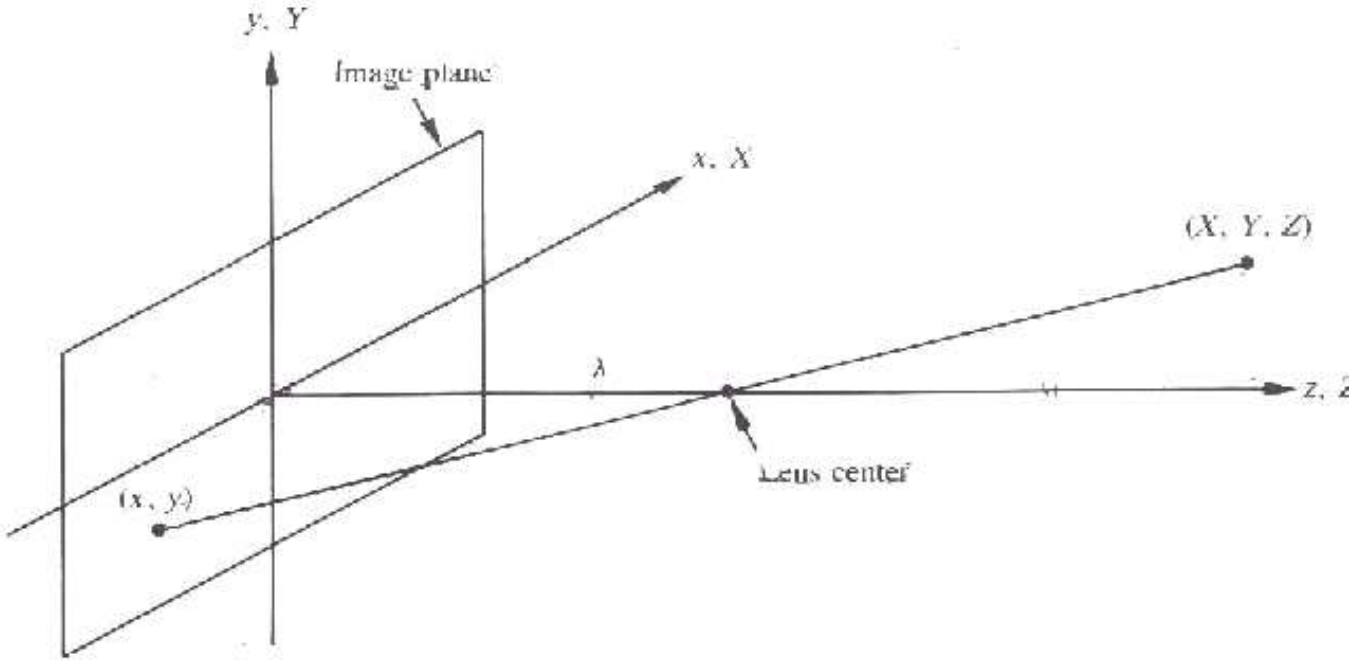


Figure 2.14 Basic model of the imaging process. The camera coordinate system (x, y, z) is aligned with the world coordinate system (X, Y, Z).

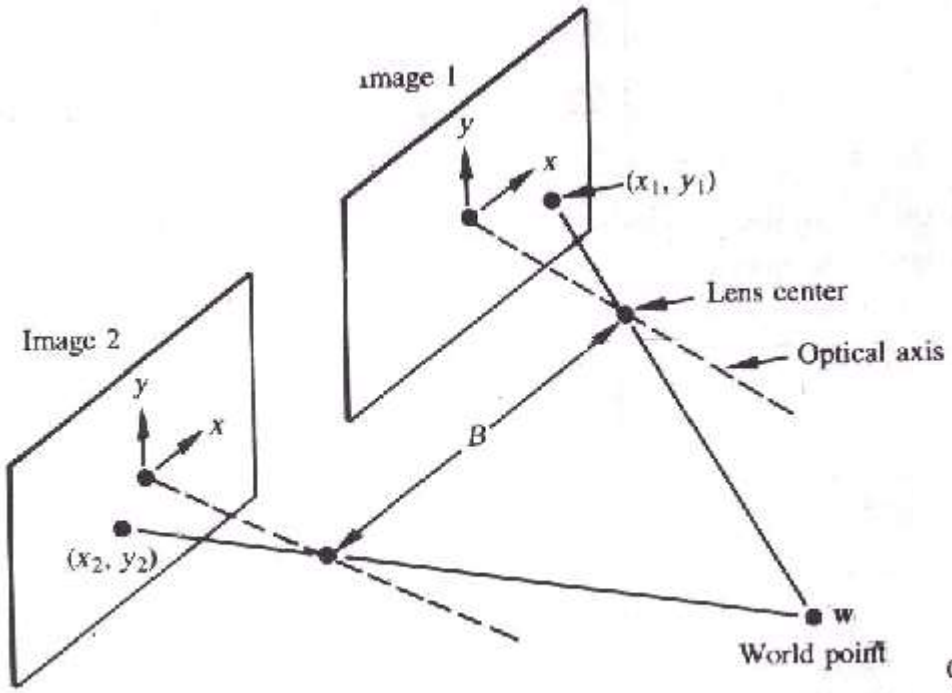


Figure 2.18 Model of the stereo imaging process. (From Fu, Gonzalez, and Lee [1987].)

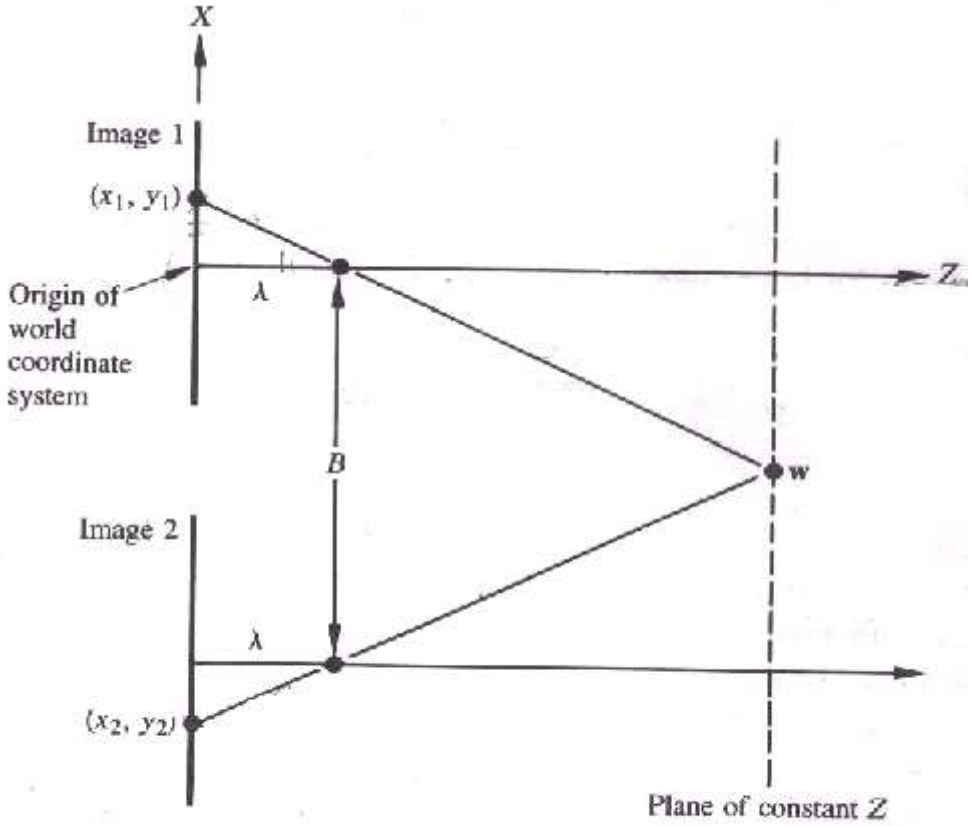


Figure 2.19 Top view of Fig. 2.18 with the first camera brought into coincidence with the world coordinate system. (From Fu, Gonzalez, and Lee [1987].)

$$-X_1 = \frac{x_1}{f}(Z_1 - f)$$

$$X_2 = \frac{-x_2}{f}(Z_2 - f)$$

$$X_2 - X_1 = B$$

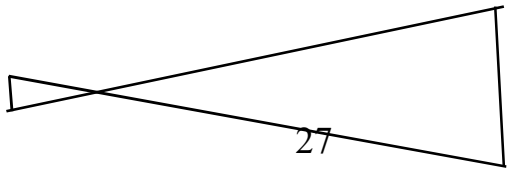
$$Z_2 = Z_1 = Z$$

$$X_2 = \frac{x_2}{f}(f - Z) = X_1 + B$$

$$Z = f - \frac{f * B}{(x_2 - x_1)} \approx \frac{f * B}{d} \quad if \quad f << B$$

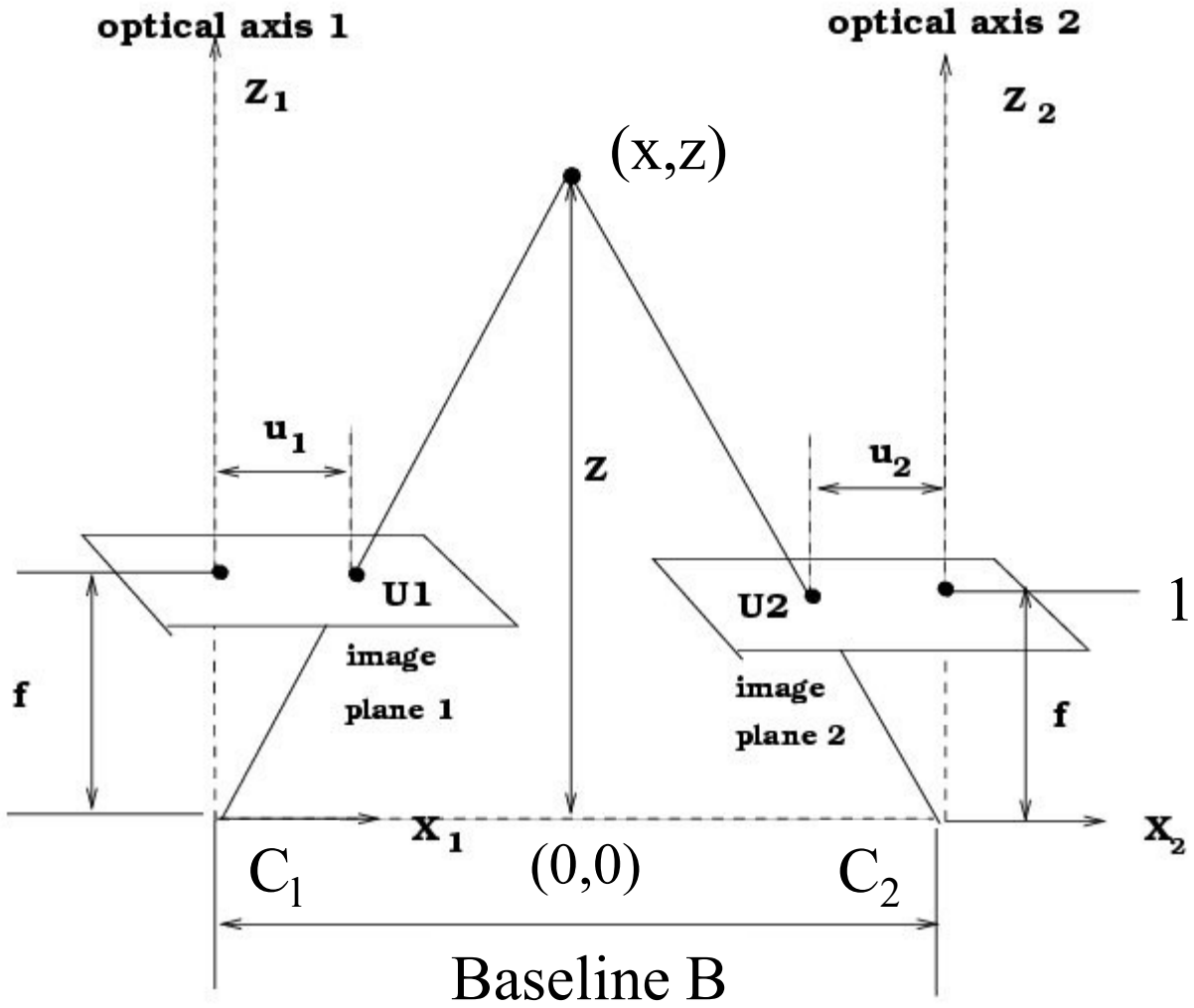
d: disparity

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



Simplest Stereo System II: $Z = \frac{f * B}{d}$

- Consider two 1D cameras with identical intrinsic parameters



1D Image Coordinate

Recovering Depth from Disparity: Theoretical Basis - Triangulation

- **Given a correspondence between two locations in the two images, the depth of a feature can be computed from the *disparity d* between the two image locations**

  - Since triangle XU_1U_2 is similar to triangle XC_1C_2, we have

$$\frac{Z-f}{Z}=\frac{B-(u_1-u_2)}{B} \quad where\ u_2<0$$

Disparity: $$d=(u_1-u_2)=f\left(\frac{B}{Z}\right)$$

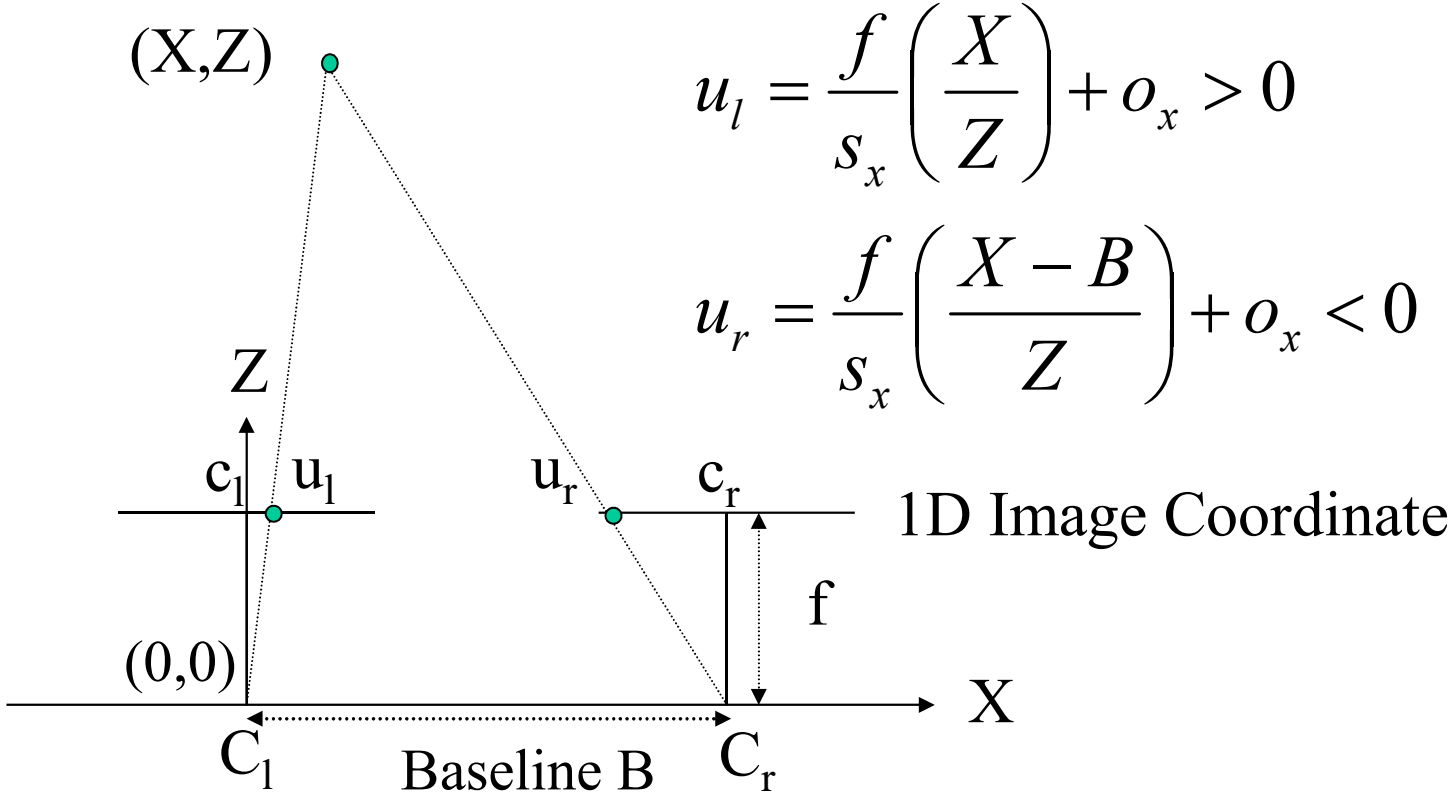
Depth: $$Z=f\left(\frac{B}{d}\right)$$

Unit-
Z: cm (or m)
B: cm
f: mm
d: pixel => um

It is clear that disparity *d* is inversely proportional to depth Z.
As depth Z approaches infinity, disparity approaches zero.

Simplest Stereo System III: $Z = \frac{f * B}{d}$

- **Consider two 1D cameras with identical intrinsic parameters**



$$u_l = \frac{f}{s_x}\left(\frac{X}{Z}\right) + o_x > 0$$

$$u_r = \frac{f}{s_x}\left(\frac{X - B}{Z}\right) + o_x < 0$$

Recovering Depth from Disparity:

- **Given a correspondence between two locations in the two images, the depth of a feature can be computed from the *disparity d* between the two image locations**

Disparity: $$d = (u_l - u_r) = \frac{f}{s_x}\left(\frac{B}{Z}\right)$$

Depth: $$Z = \frac{f}{s_x}\left(\frac{B}{d}\right)$$

Unit-
Z: cm (or m)
B: cm
f: mm
d: pixel => um

Difficulty to Design Stereo

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

- **At least 2 variants by multiplication factor**

The Ambiguity of the Correspondence Problem

- **The goal of correspondence algorithms is to find matching locations in the left and right images**
- **Correlation Based Approaches:**
  - A common approach to find correspondences is to search for local regions *w* that appear similar
  - Algorithm
    - » For each u_l, find a d that minimizes/maximizes $c(u_l, d)$

cost function

$$c(u_l, d) = \psi(I_l(u_l - w{:}u_l + w), I_r((u_l - d) - w{:}(u_l - d) + w)$$

- **Correlation Approaches**
  1) Note that, in locations of low variance, it is difficult to find unique matches
  2) Strictly speaking, correlation approaches are only correct if the surface being viewed is front-parallel

Correspondence (Match)

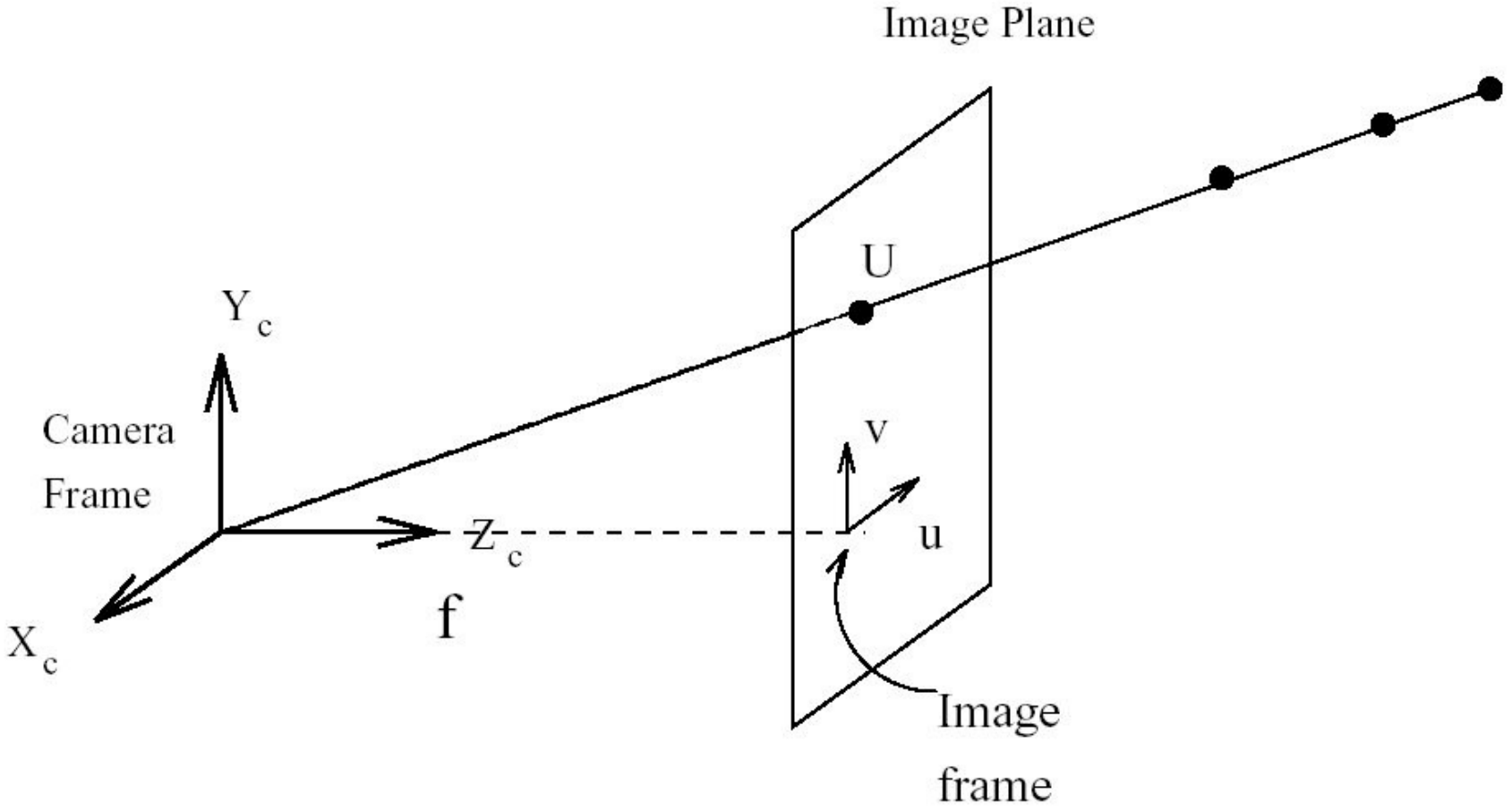
- **Invariant by perspective transformation**
  1) The pixel
  2) The edge pixel: ~~The feature (feature based approach)~~
     - » Instead of trying to match every single pixel in the left image to its mate in the right image, we could simply extract ~~features such as~~ edges in the left images and look for matches for those features only.
  3) The image regions (J: feature-based approach):
     - » Depending on how these regions are extracted, their shapes and the intensity-based feature attached to them may or may not be invariant by perspective transformation.
- **Since stereo matching is an ill-posed problem, global constraints are often employed**
- **These constraints effectively represent prior knowledge about the scenes**

Find Corresponding Point

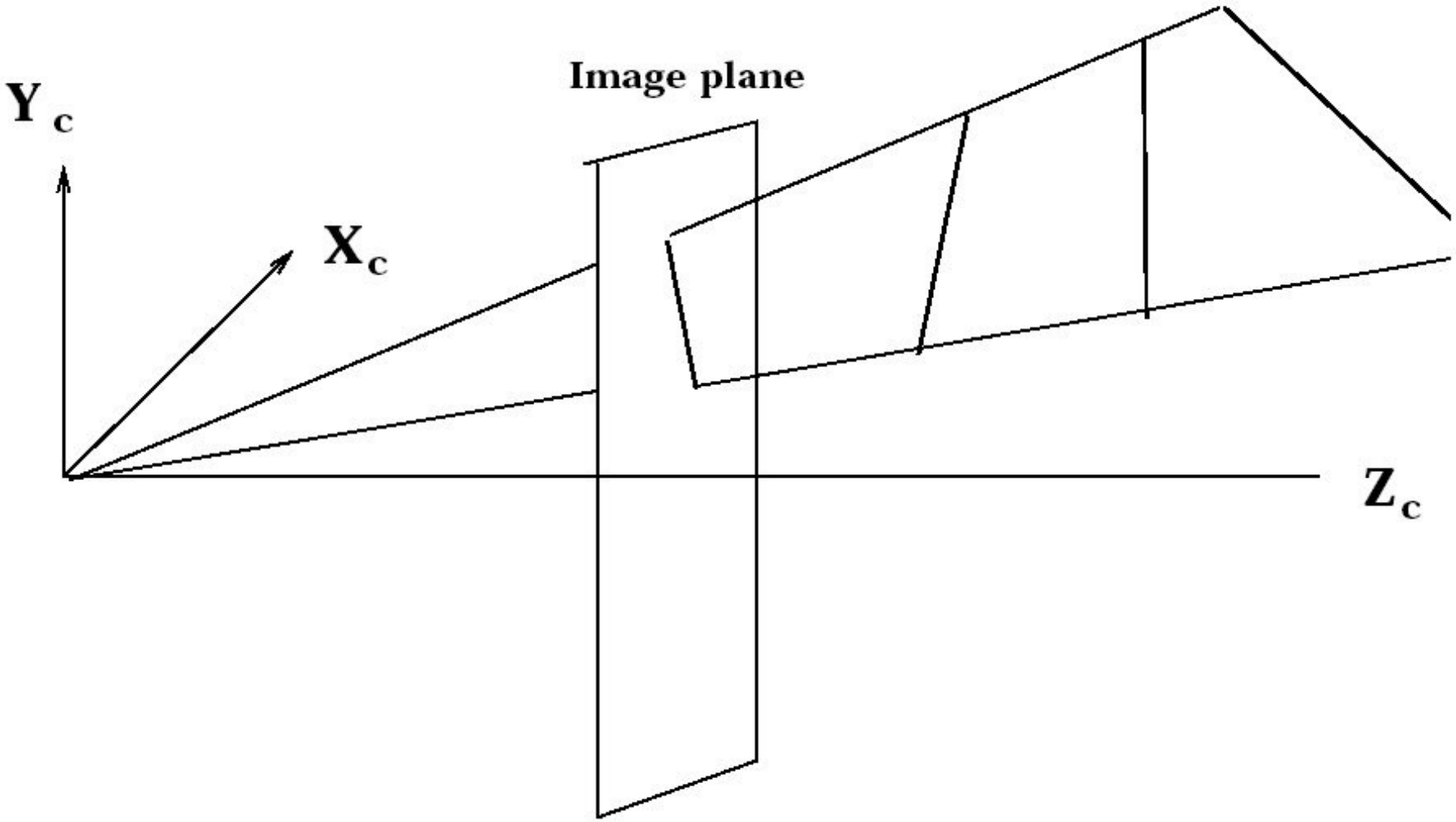
- **Find corresponding points:**
  1) **Search range: Along one direction, such as x-axis.**
     - Epipolar line: Reduce from 2D to 1D
  2) **Match window size**
     - **(1) Pixel**
     - **(2) Edge**
     - **(3) Region/feature**
  3) **Match method (similarity measure)**

Ill-Posed Problem

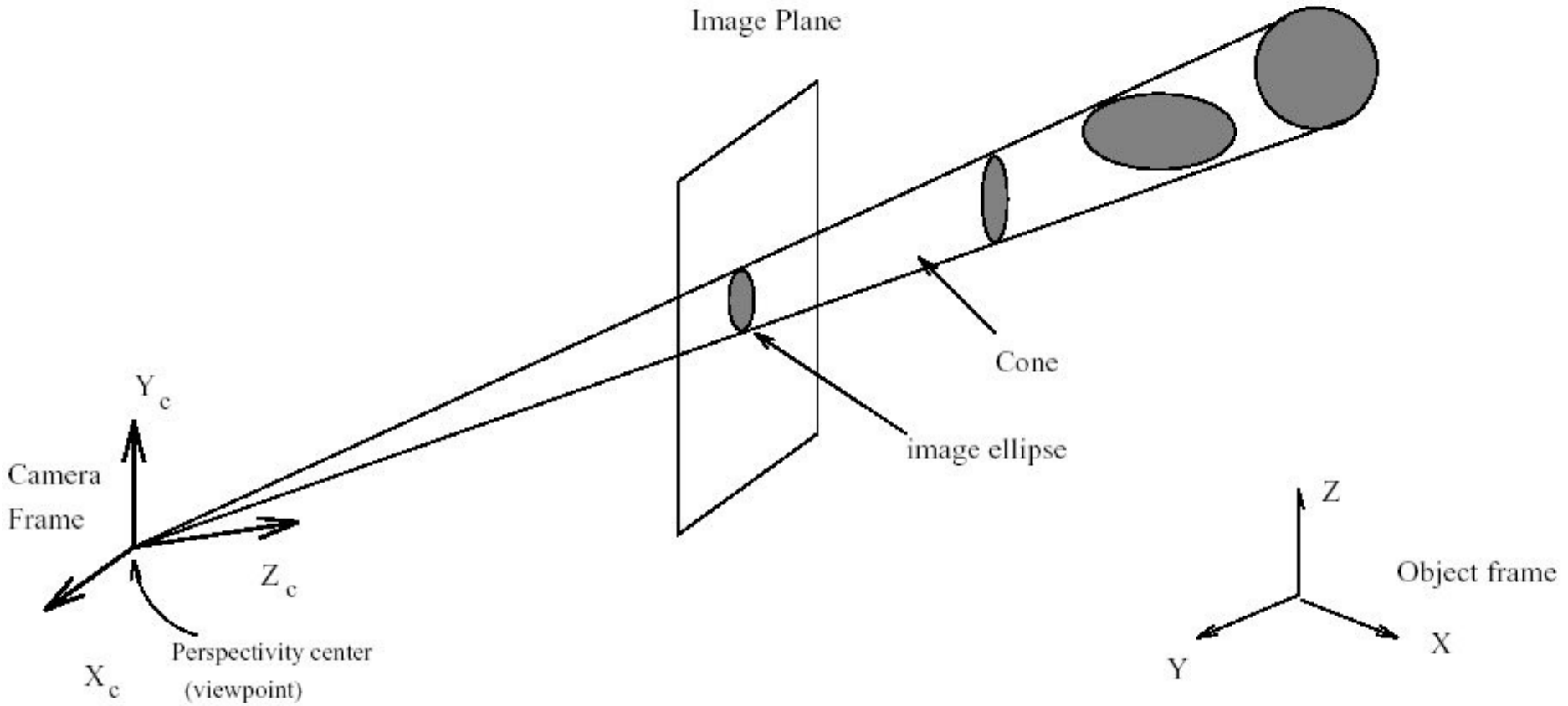
An Ill-posed Problem: Point



An Ill-posed Problem: Line

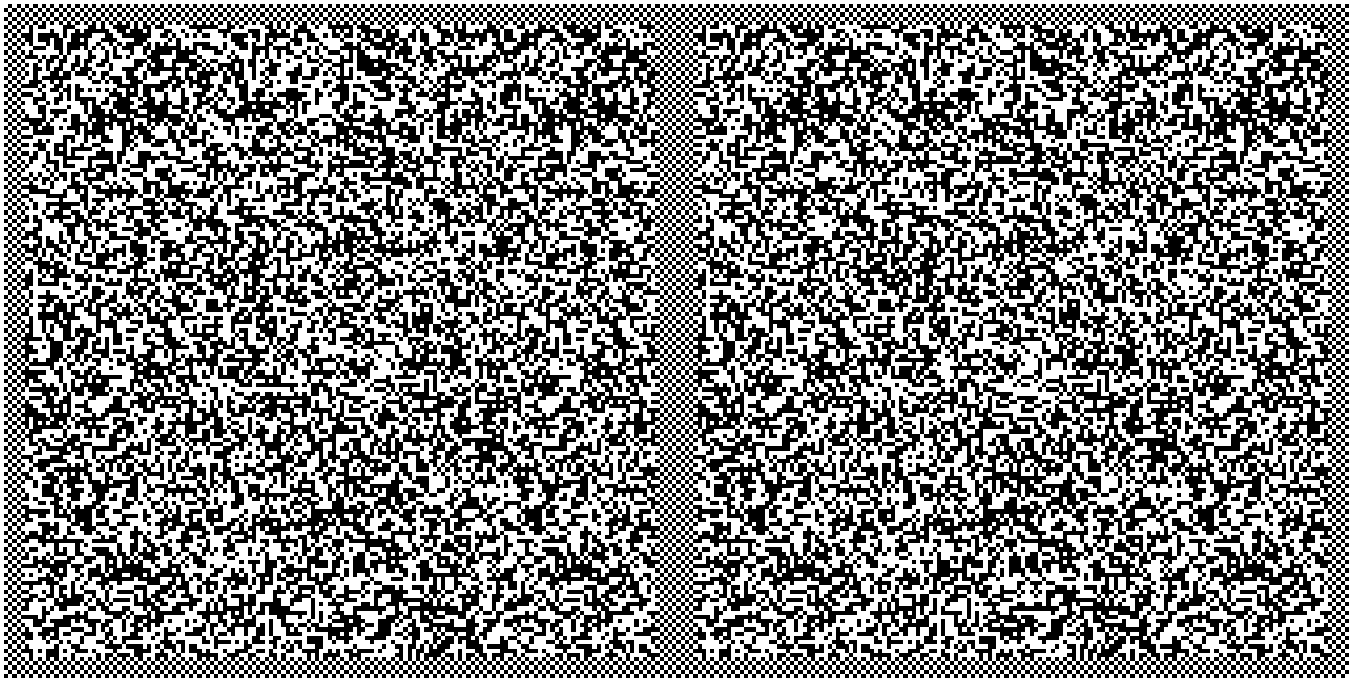


An Ill-posed Problem: Curves



❑ Features vs. Pixels?

- ➢ Do we extract features prior to matching?



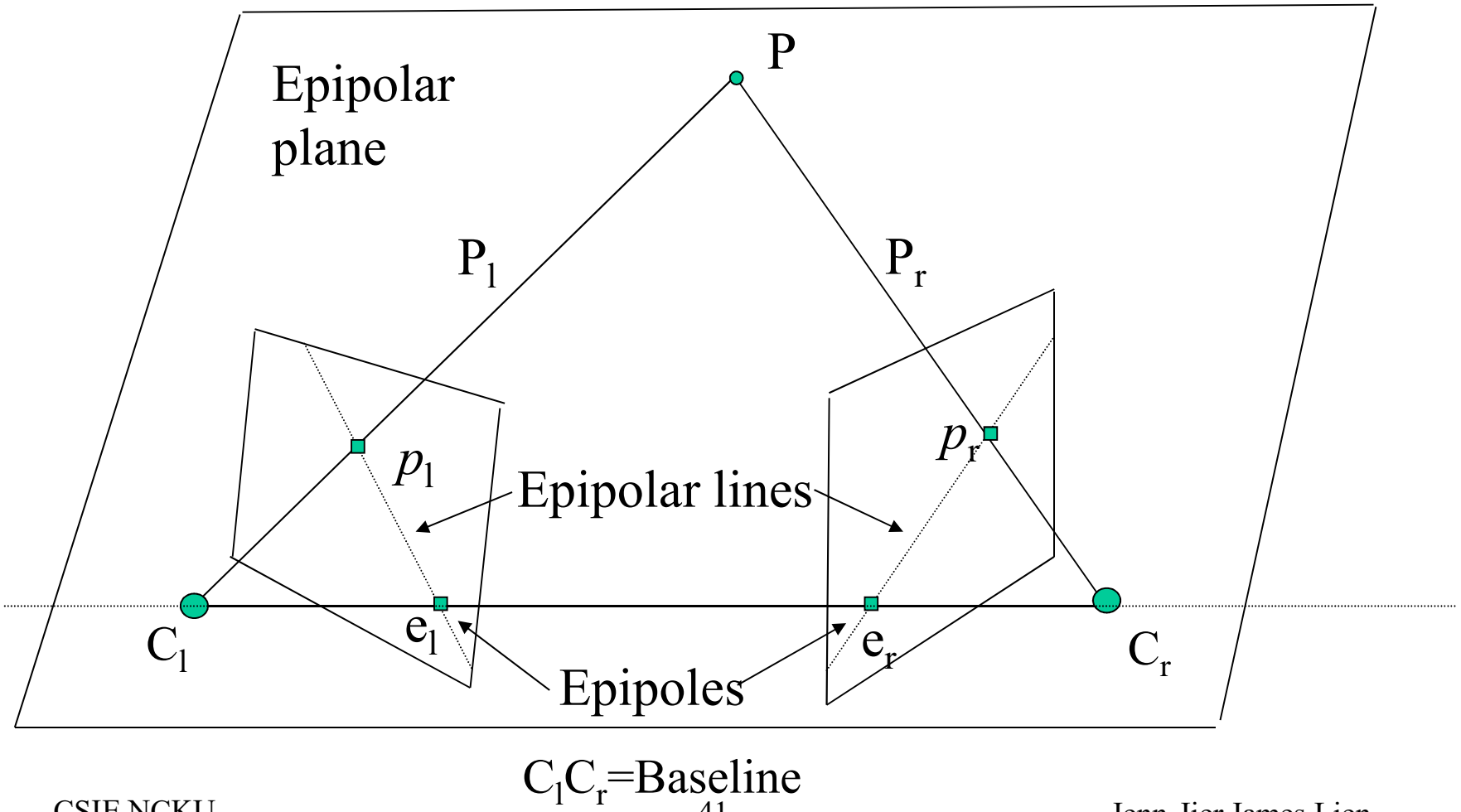
Julesz-style Random Dot Stereogram

Constraints and Matching Techniques

- **Constraints: Each of these constraints can be used to reduce the ambiguity of correspondence:**
  1) The epipolar constraint: 2D search to 1D search
  2) Uniqueness of match
  3) Continuity
  4) The ordering constraint
  5) The disparity gradient constraint
  6) The geometric constraints
- **Matching Techniques:**
  1) Dynamic Programming
  2) Relaxation labeling
  3) Graphcuts

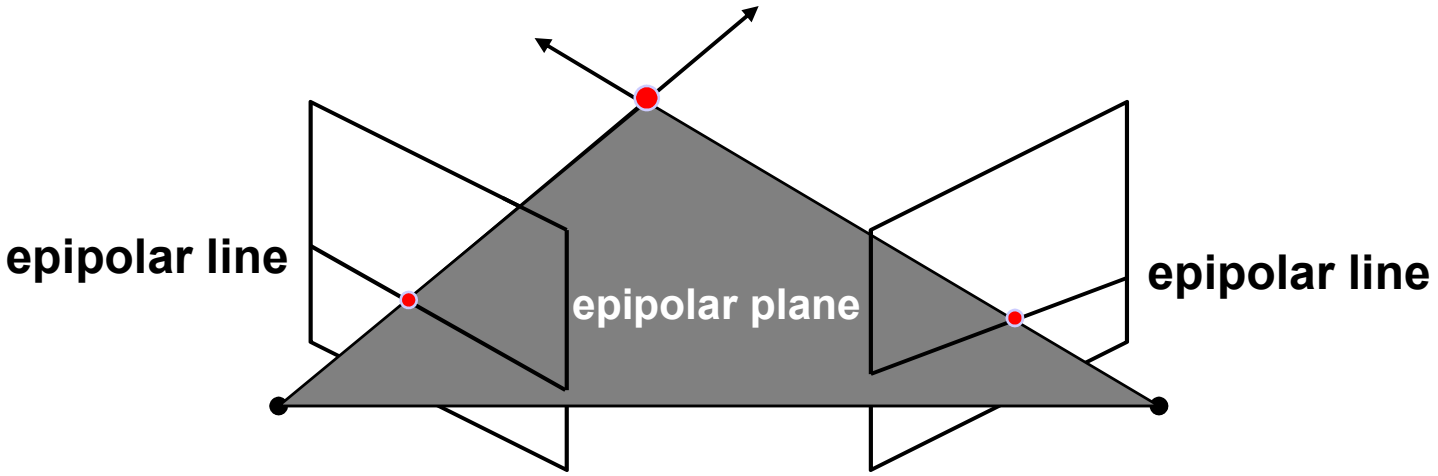
The Epipolar Constraint: 2D Search to 1D Search - Epipolar Geometry

- **The epipolar constraint reduces correspondence problem from 2D to 1D search along *conjugate epipolar lines***



Stereo correspondence:

- **Determine Pixel Correspondence**
  - Pairs of points that correspond to same scene point



- **Epipolar Constraint**
  - Reduces correspondence problem from 2D to 1D search along *conjugate epipolar lines*
  - Java demo: http://www.ai.sri.com/~luong/research/Meta3DViewer/EpipolarGeo.html

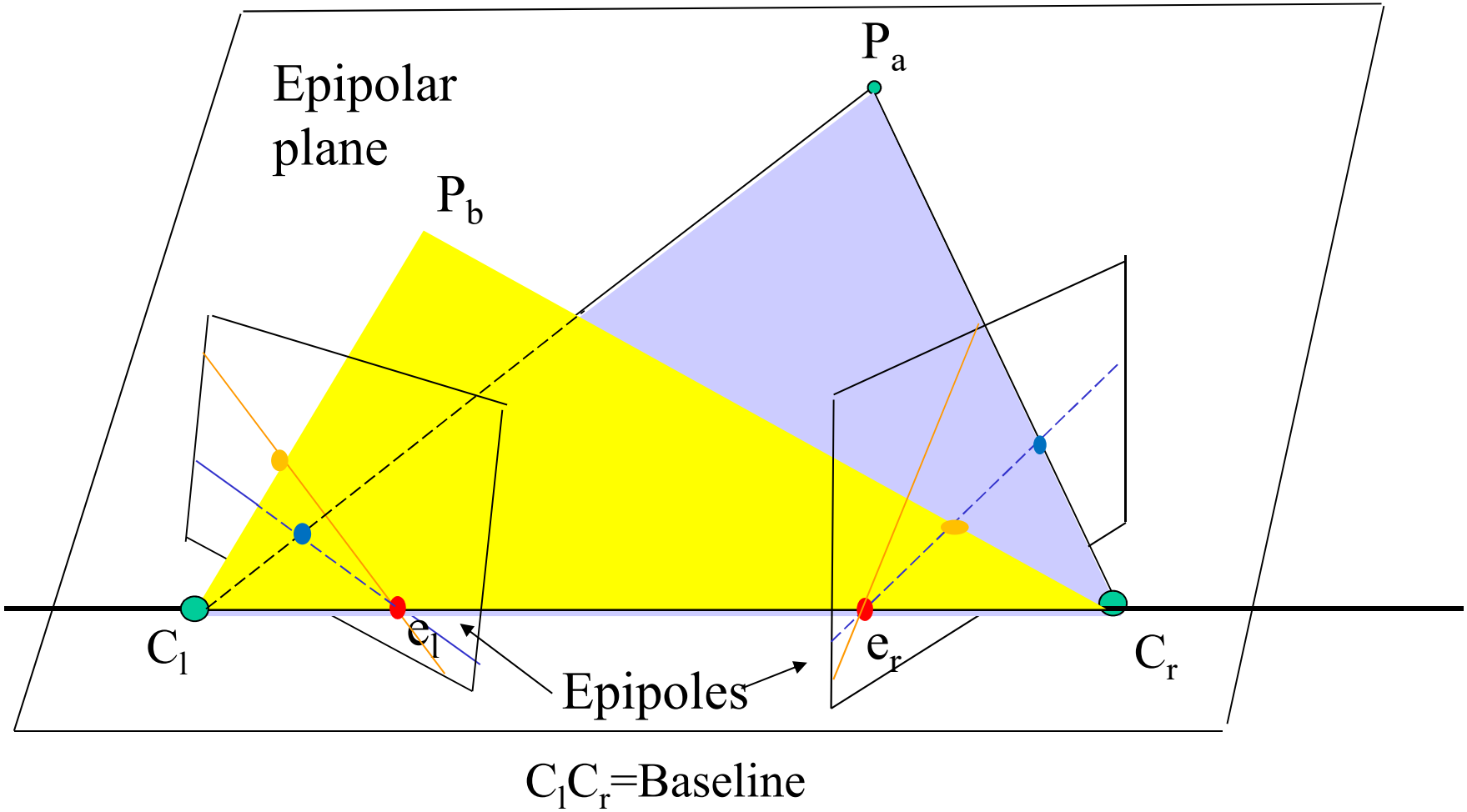
Epipolar Geometry (Perspective Projection):

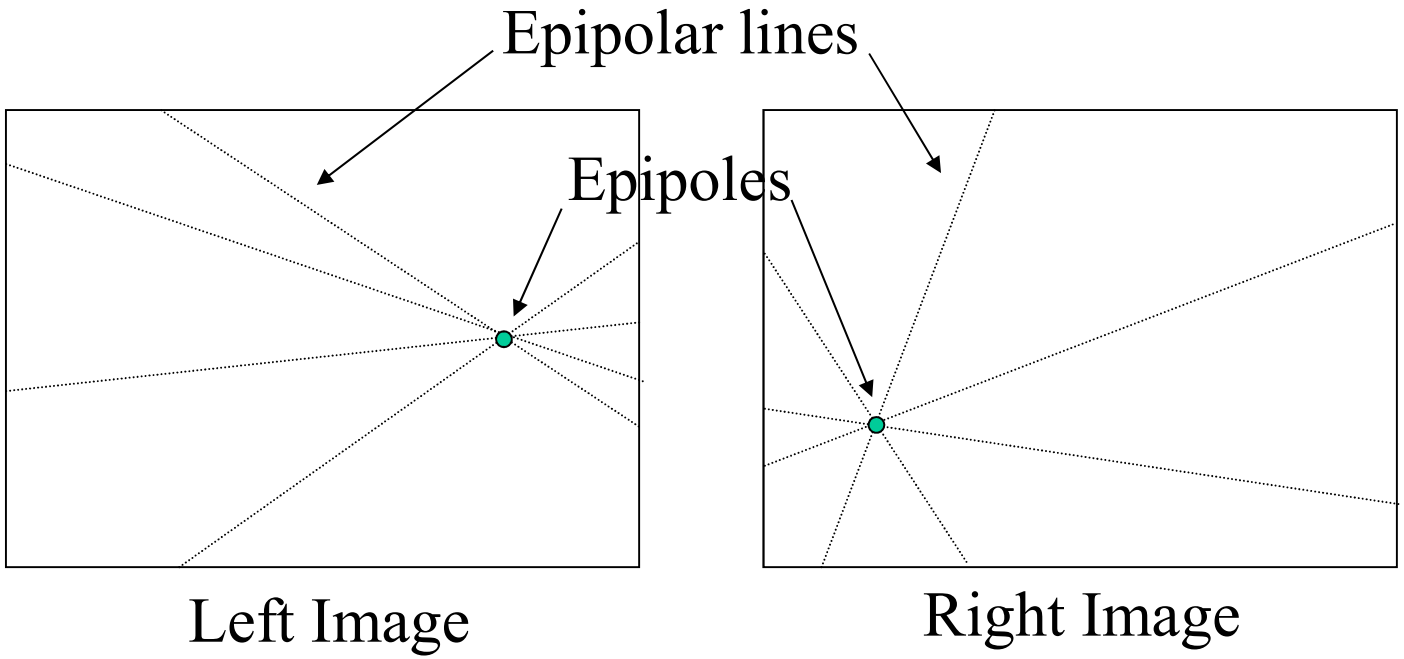
- Consider the set of planes that pass through the centers, C_l and C_r, of projection of both cameras. These planes are referred to as *epipolar planes*
- In each image, the points, p_l or p_r, on a given epipolar plane project onto an *epipolar line*.
- The *epipole*, e_l (or e_r), in each image corresponds to the projection of the center, C_r (or C_l), of projection of the other camera.
- All of the epipolar lines in each image pass through the epipole in that image.

The Epipolar Constraint

- **The epipolar constraint:**
  - For every point, such as p_l, observed in the left image, we know that its correspondent, such as p_r, must lie along the corresponding epipolar line in the right image
  - For every epipolar line in the left image, there is a corresponding epipolar line in the right image
- **The epipolar constraint: Reduces correspondence problem from 2D to 1D search along *conjugate epipolar lines***
  - That is, for each epipolar line in the left image locates the corresponding epipolar line in the right image and then finds matches using a 1D stereo algorithm
- **This observation can substantially simplify the search for correspondences**

- **All of the epipolar lines in each image pass through the epipole in that image**




Epipolar lines
Epipoles
Left Image
Right Image

Geometry of Epipoles

- **There are three cases for the geometry of the epipoles:**

  1) Case I - Fig. 6.2: Both epipoles are at a finite distance in their respective focal planes

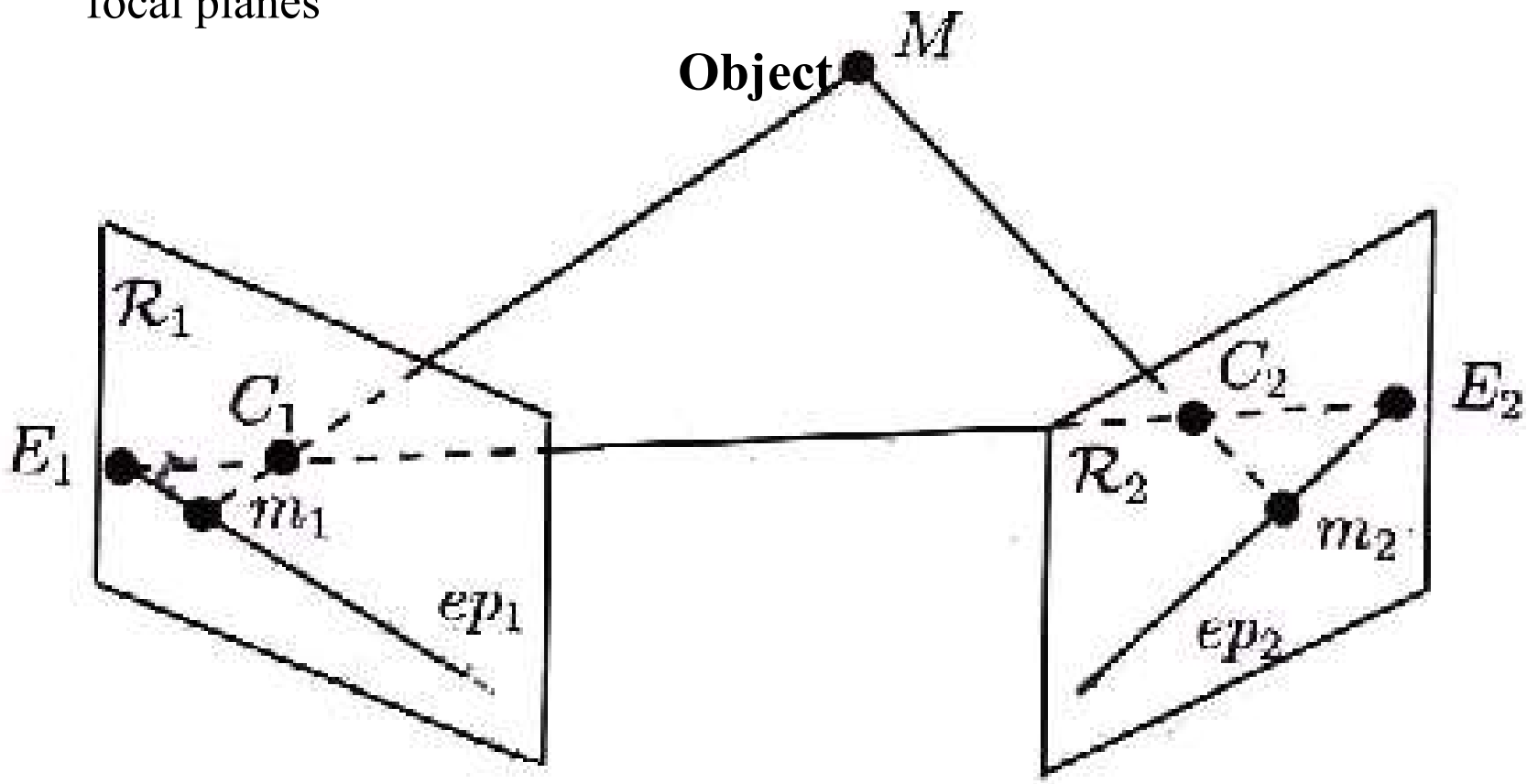


Figure 6.2 The epipolar geometry.

2) Fig. 6.3: One is at a finite distance, and the other is at infinity

Object

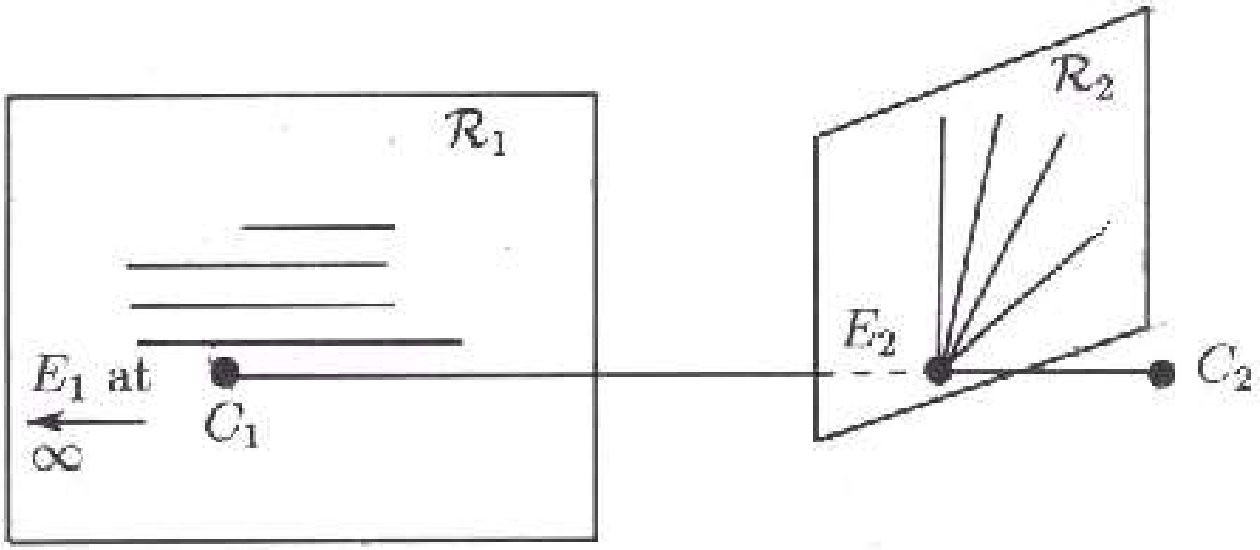


Figure 6.3 $\langle C_1, C_2 \rangle$ is parallel to the plane $\mathcal{R}_1$: E_1 is at ∞; the epipolar lines are parallel in the plane $\mathcal{R}_1$ and intersect at E_2 in the plane $\mathcal{R}_2$.

3) Fig. 6.4: They are both at infinity

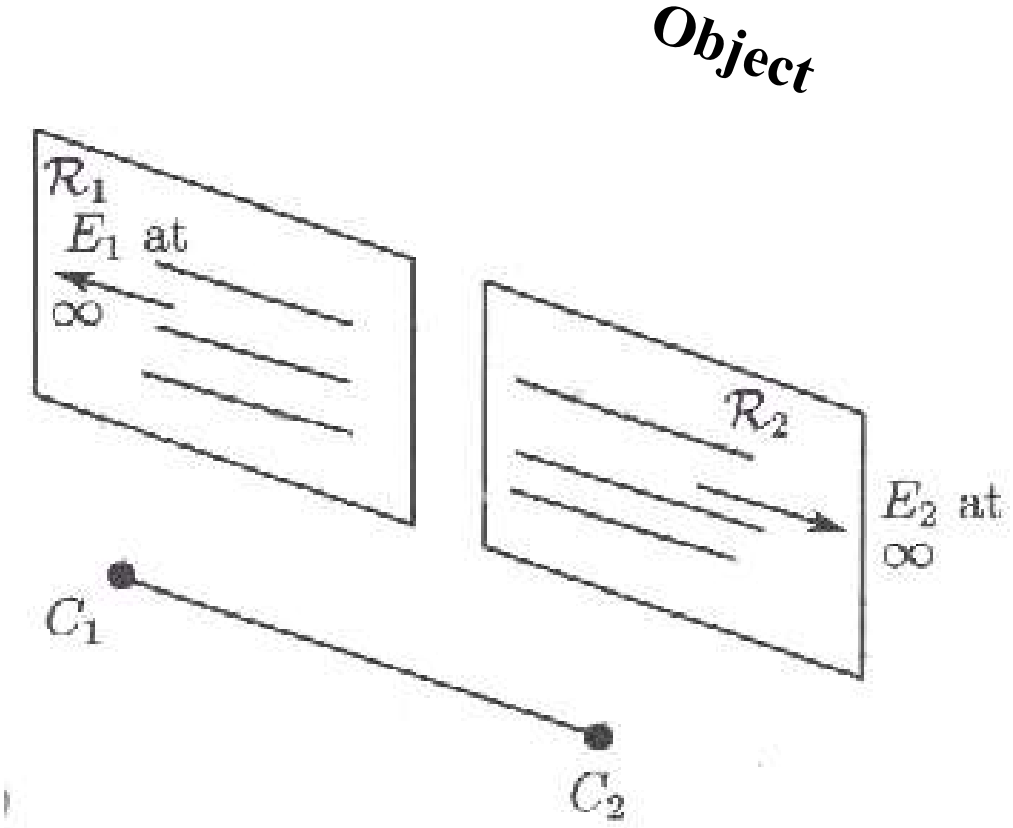
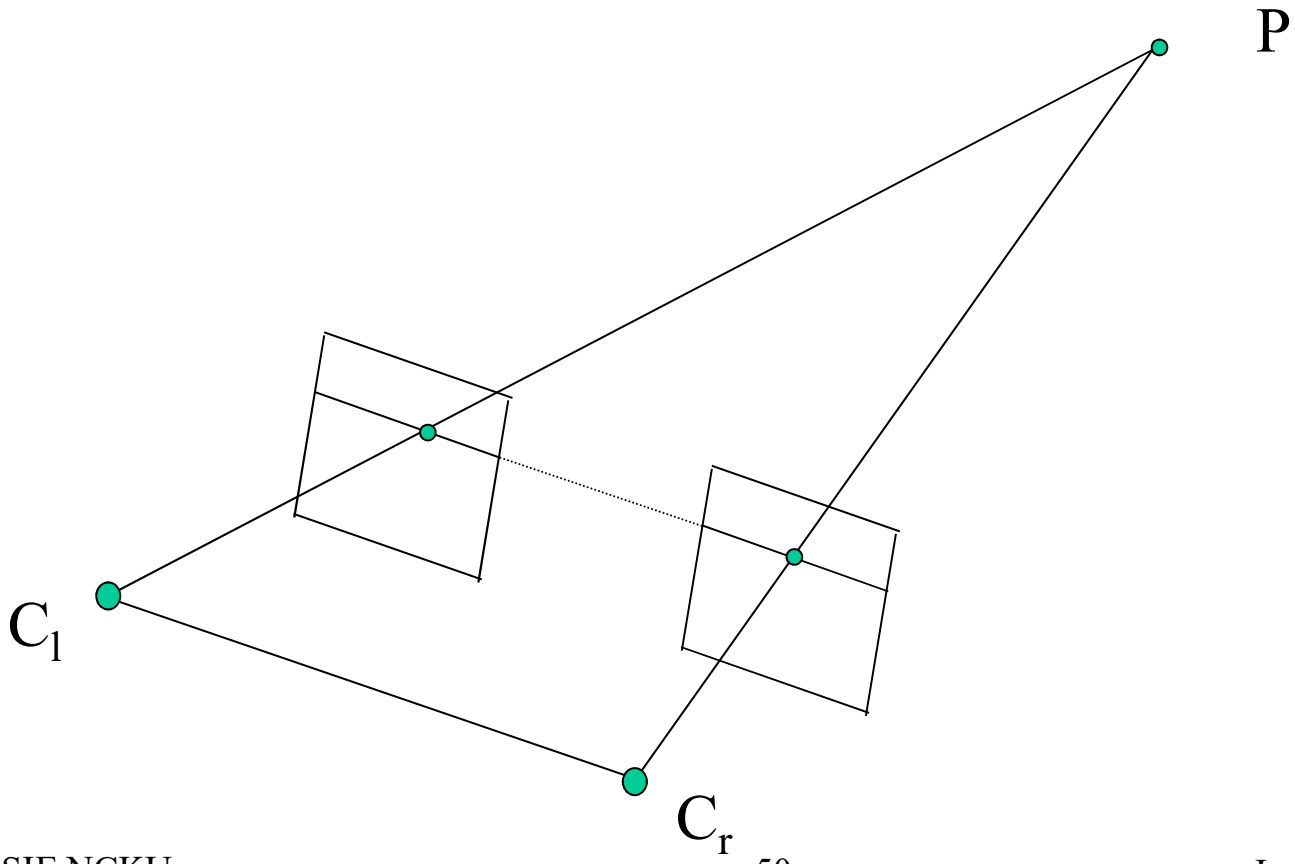
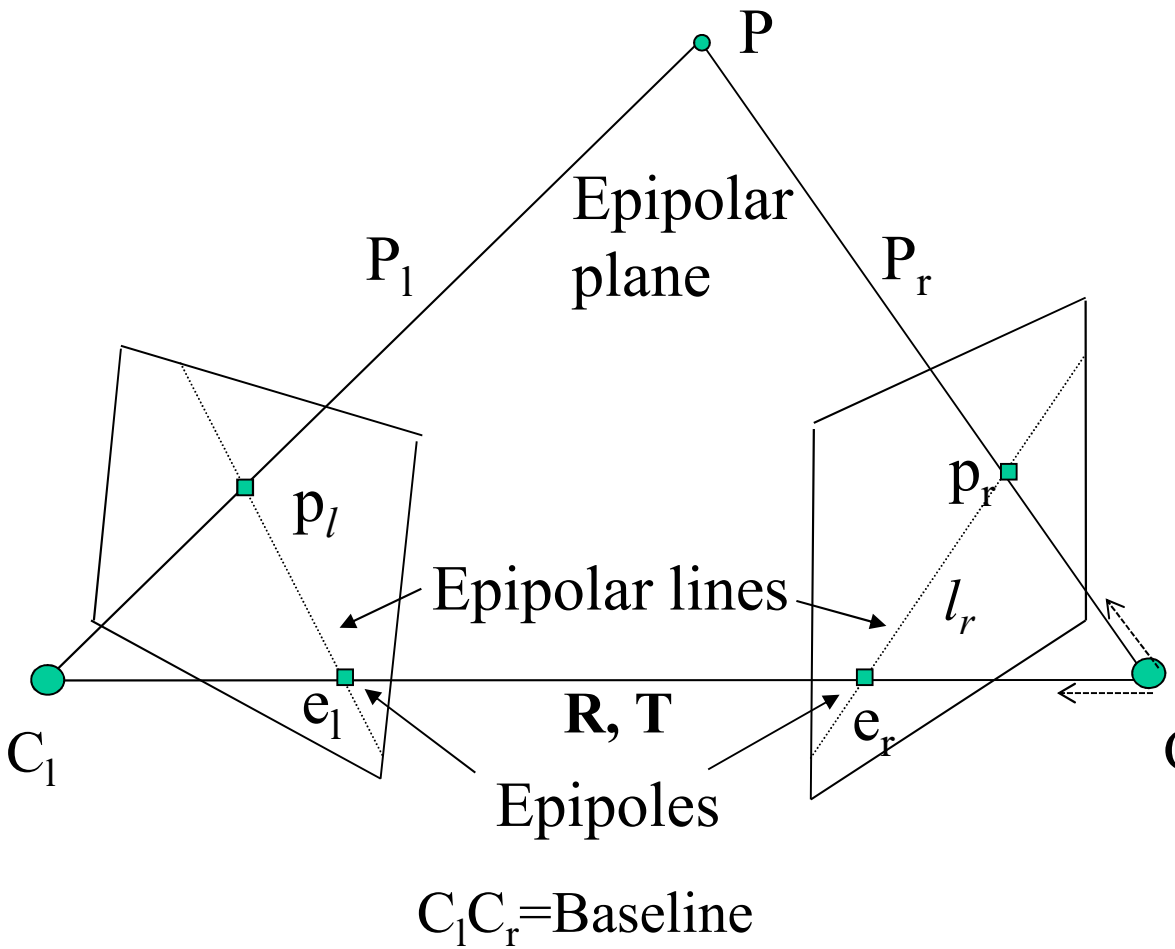


Figure 6.4 $\langle C_1, C_2 \rangle$ is parallel to the planes $\mathcal{R}_1$ and $\mathcal{R}_2$: E_1 and E_2 are at ∞; the epipolar lines are parallel in both planes $\mathcal{R}_1$ and $\mathcal{R}_2$.

- **In the special case of the stereo setup shown above where the image planes are aligned with each other, the epipolar lines correspond to rows in the image**
- **That is the epipoles in both images are at infinity along the x axis.**



Computing the Epipolar Geometry



$$p_l = A_l P_l$$

$$p_r = A_r P_r$$

$$P_r = R(P_l - T)$$

$\mathbf{e_r} \times \mathbf{p_r} = \mathbf{l_r}$

Normal direction of epipolar line lr

$=(a,b,c)$

$ax+by+c=0$

$(a,b,c).(x,y,1)=0$

$\|n_{lr}\|=1$

Essential matrix

P_r $\quad$ 90^0

$$(P_l - T)^T \; T \times P_l = (P_l - T)^T J(T) P_l = 0$$

$$\Rightarrow (R^T P_r)^T J(T) P_l = 0$$

$$P_r^T R J(T) P_l = 0$$

$$P_r^T E P_l = 0$$

$$E = RJ(T)$$

- **The matrix E = RJ(T) is referred to as the** Essential Matrix

Fundamental Matrix

- From the previous equations we can readily conclude that:

$$P_r^T E P_l = 0$$

$$\Rightarrow (A_r^{-1} p_r)^T E (A_l^{-1} p_l) = 0$$

$$p_r{}^T A_r^{-T} E A_l^{-1} p_l = 0$$

$$\boxed{\begin{aligned} p_r{}^T F p_l &= 0 \\ F &= A_r^{-T} E A_l^{-1} \end{aligned}}$$

- The matrix F is referred to as the Fundamental matrix
- This matrix can be interpreted as the matrix which maps **points** in the left image onto the corresponding **epipolar line** in the right image

- **Rewriting F**

$$F = A_r^{-T} E A_l^{-1} = A_r^{-T} R J(T) A_l^{-1}$$

$$= A_r^{-T} R A_l^T A_l^{-T} J(T) A_l^{-1} \propto (A_r^{-T} R A_l^T) J(A_l T)$$

$$\propto H_\infty J(e_l)$$

$$H_\infty \propto A_r^{-T} R A_l^T$$

$$e_l \propto A_l T$$

$\underline{\mathbf{H}_{\mathbf{inf}}}$ (H_∞)

- **H is referred to as the homography of the plane at infinity. It can be seen as a mapping between the projections of points at infinity in the left image and their correspondents in the right image. (so translation factor is gone ?) It can also be seen as the mapping between epipolar lines in the two images**
- **Properties of H_{inf}**
  - H_{inf}^T maps the projections of points at infinity in the right image onto their correspondents in the left image
  - H_{inf}^T maps the epipole in the right image onto the epipole in the left image
  - H_{inf} maps epipolar lines in the left image onto their correspondents in the right image

Mapping of Epipolar Lines

$$p_r^T F p_l = 0$$

$$p_r^T H_\infty J(e_l) p_l = p_r^T H_\infty (e_l \times p_l) = p_r^T H_\infty l_l = 0$$

$$\Rightarrow l_r \propto H_\infty l_l$$

Epipolar lines

- **N.B. the set of epipolar lines in an image can be thought of as a projective space of dimension 1, a projective subspace of the set of all lines in RP^2**

Points at Infinity (Mapping of Epipoles)

- **Consider a point at infinity (so translation factor is gone ?) whose homogenous coordinates wrt the left camera frame are (v 0). Let p_l and p_r denote the projection of this point onto the left and right cameras respectively**

$$p_l \propto A_l v$$

$$p_r \propto A_r R v$$

$$p_l \propto A_l R^T A_r^{-1} p_r \propto H_\infty^T p_r$$

Recovering F from Point Correspondences

- **Given eight or more point correspondences between the two images, the fundamental matrix, F, can be recovered from a set of homogenous linear equations using SVD**

Analyzing F

- **The epipoles in the left and right images can be recovered by finding the kernels of F and F^T respectively**

Recovering H_{inf}

- **H_{inf} cannot be recovered from point correspondences alone, some other information such as the correspondences between points at infinity must be supplied**

Recovering E

- **If the matrices of intrinsic parameters are known, it is a simple matter to recover the essential matrix, E (up to a scale) from the Fundamental matrix, F**

$$E \propto A_r^T F A_l$$

Recovering R and T

- **If the essential matrix is known (up to a scale) it is possible to recover the rotation matrix, R and the translation vector T (up to a scale)**
- **The translation vector T can be recovered *up to an unknown scale factor* by finding the null space of the vector E since ET=0**

$$E = RJ(T)$$

$$E^T E = (J(T))^T J(T) = -J(T)^2$$

$$tr(E^T E) = 2\|T\|^2$$

$$\hat{E} = \frac{1}{\|T\|} E = RJ(\hat{T}); \quad \hat{T} = \frac{1}{\|T\|} T$$

- **The rotation matrix R can be recovered from E using the following trick.**
- **Note that there will be four possible solutions for R and T corresponding to different choices of sign for E and T. We can disambiguate between the solutions by choosing the one which corresponds to having all of the recovered points in front of both cameras**

$$\hat{E}J(\hat{T}) = (RJ(\hat{T}))J(\hat{T}) = \begin{pmatrix} R_1^T \\ R_2^T \\ R_3^T \end{pmatrix} J(\hat{T})^2 = \begin{pmatrix} w_1^T \\ w_2^T \\ w_3^T \end{pmatrix}$$

$$R_1 = w_1 + w_2 \times w_3$$

$$R_2 = w_2 + w_3 \times w_1$$

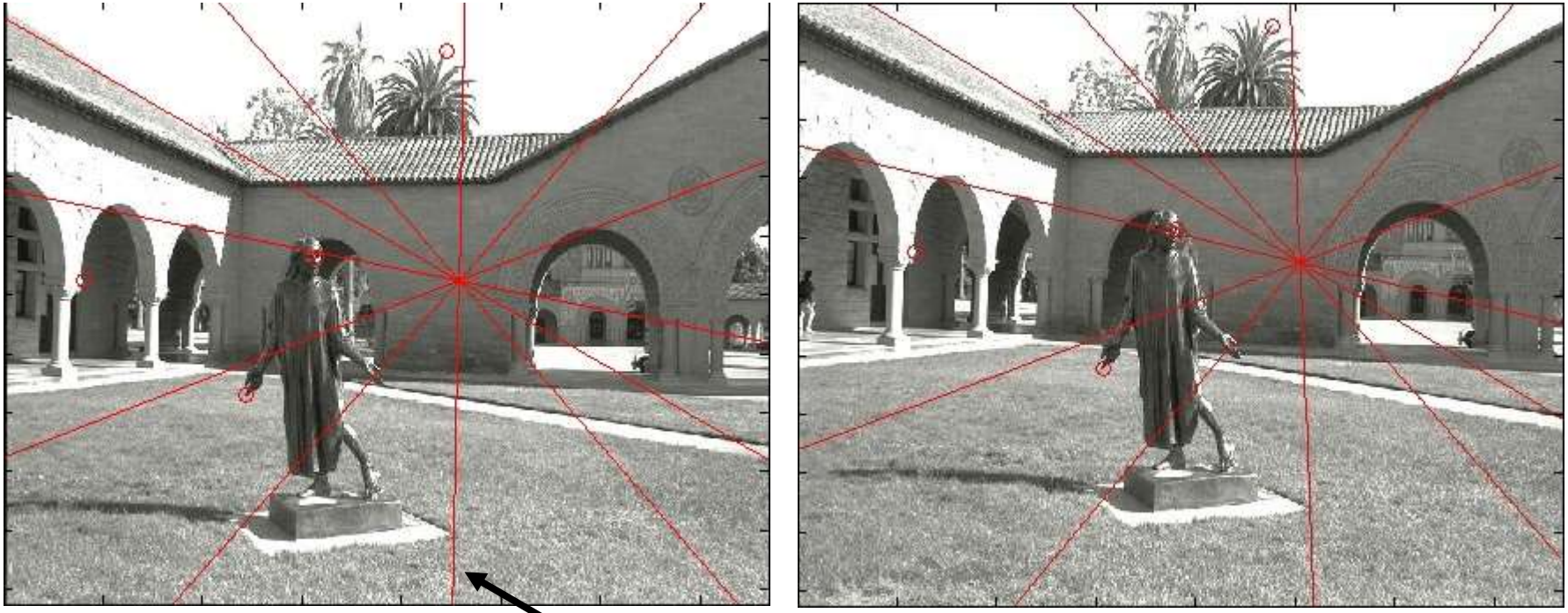
$$R_3 = w_3 + w_1 \times w_2$$

Reconstructing Points (3D Points ?)

- **Consider two cameras with projection matrices M_l and M_r and a point (3D point ?) P that projects into both of them. We can recover P from its projections quite simply.**

Projective Reconstruction

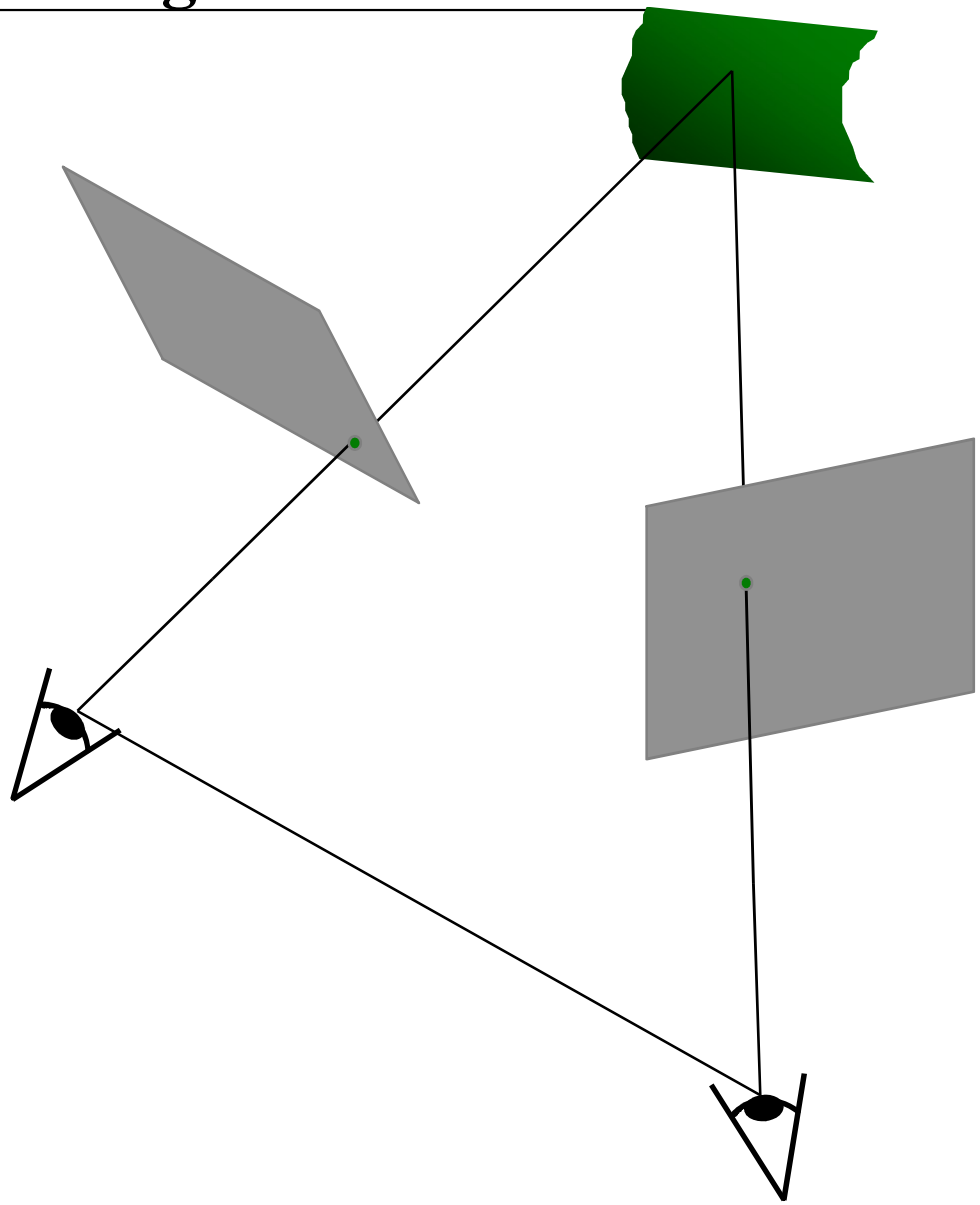
Epipolar Geometry and Fundamental Matrix: Apply to Panorama



epipolar line
(epipole: intersection of all epipolar lines)

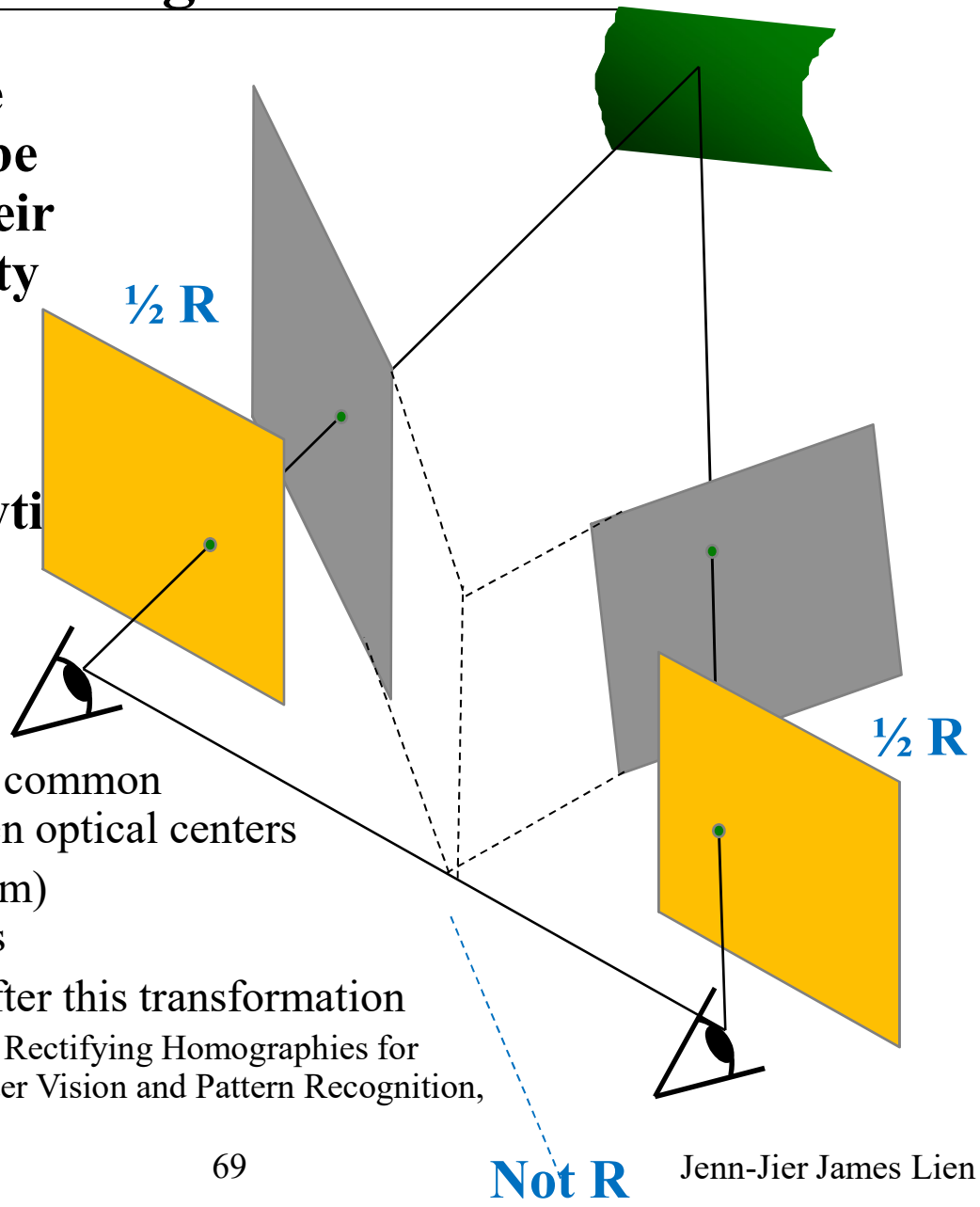
- **Can apply to panorama**

Stereo Image Rectification



Stereo Image Rectification

❑**For computational convenience, the two image planes are often chosen to be coplanar and parallel to their baseline (this means equality in focus length). Such an arrangement can be accomplished either physically or through analyti transformation**



❑ **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.

- **Given the extrinsic and intrinsic stereo parameters, we can compute the rotation matrices needed to rotate the cameras such that the conjugate epipolar lines are collinear and are parallel to the baseline. The new image coordinates are obtained by projecting the original coordinates on the new image plane.**
- **The new image plane is found by imposing two conditions**
  - It is parallel the baseline
  - It is parallel to the intersection line of the left image plane and right image plane.

  The distance of the new image plane to the baseline can be adjusted as a scale factor so that the new images fit to the sizes of the original images.

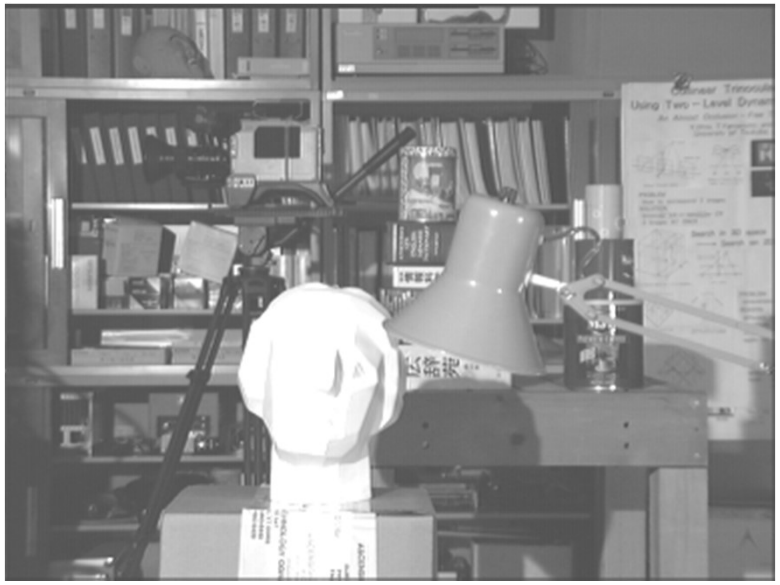
Stereo Matching Algorithms

- **Possible Candidates for Matching:**
  1) Pixel
  2) Edge pixel ~~(feature)~~
  3) Image region

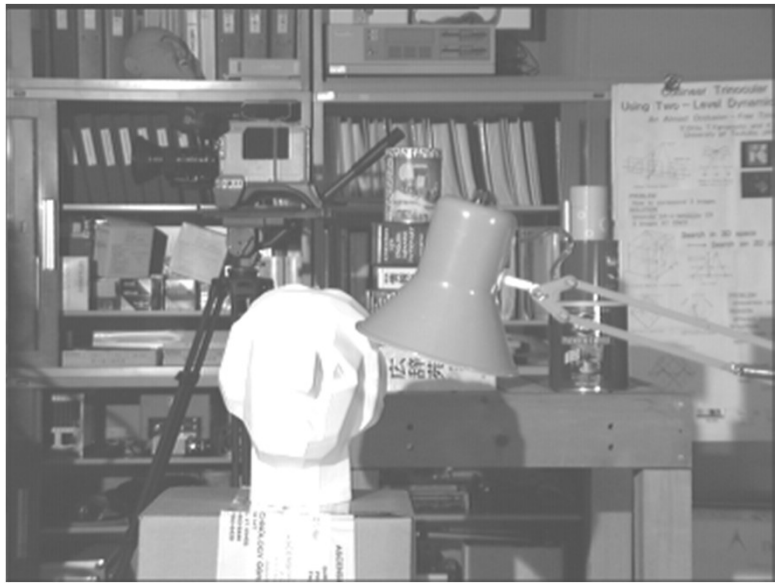
- **Match Pixels in Conjugate Epipolar Lines**
  - Assume brightness constancy
  - This is a tough problem
  - Numerous approaches
    - » dynamic programming
    - » smoothness functional
    - » more images (triocular, N-ocular)
    - » graph cuts

http://vision.middlebury.edu/stereo/data/

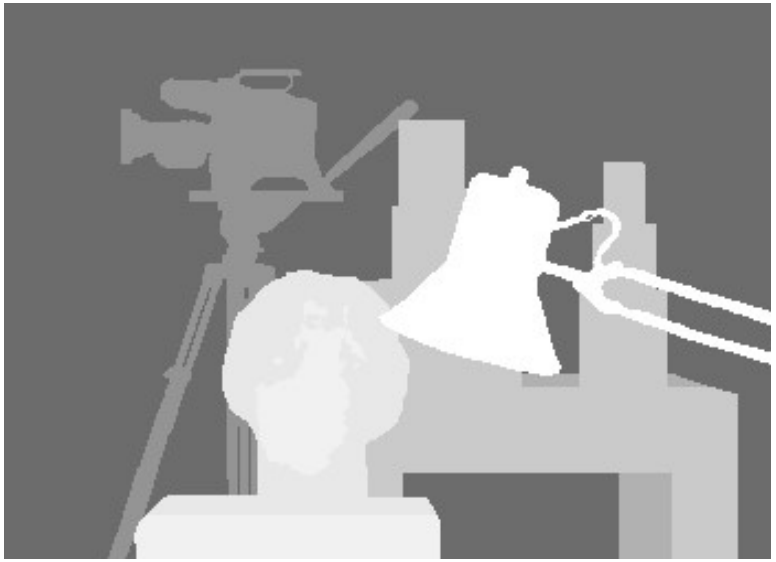
Binocular Stereo



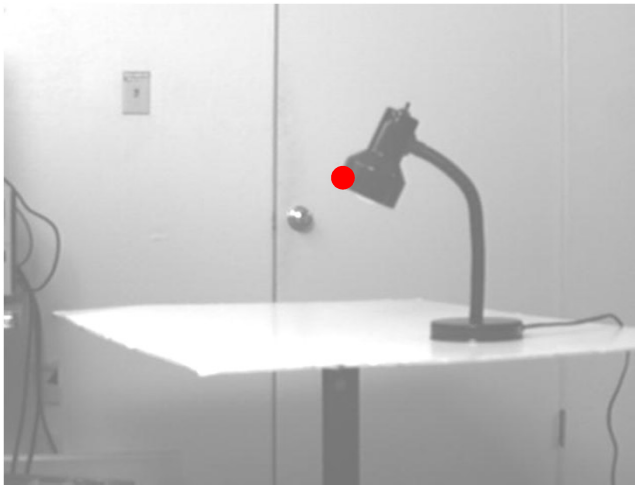
left



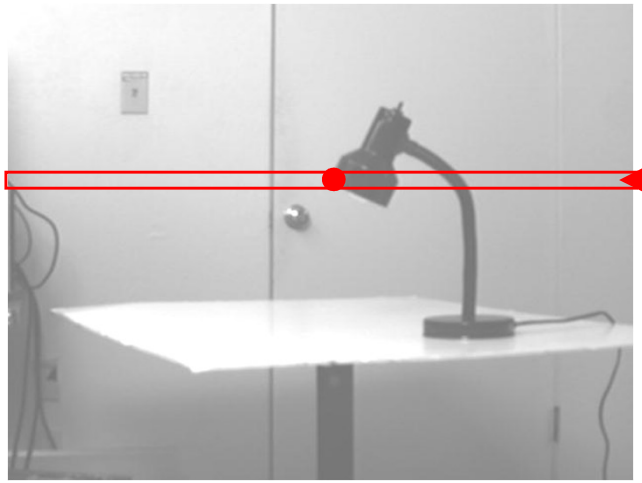
right



Binocular Stereo:

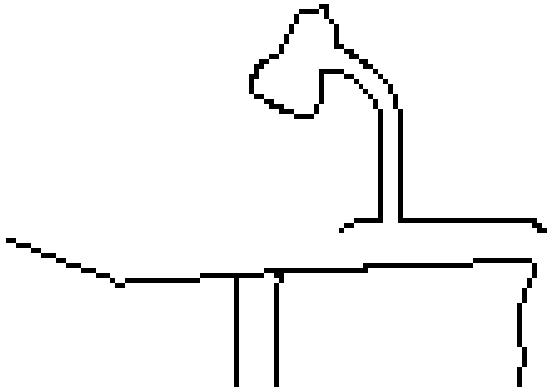


left



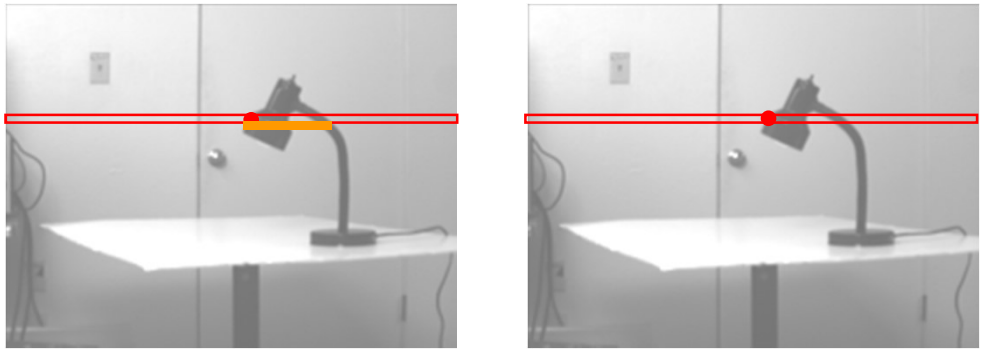
right

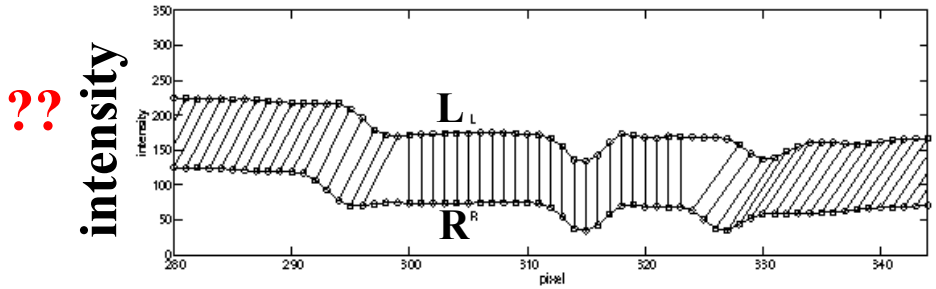
disparity map

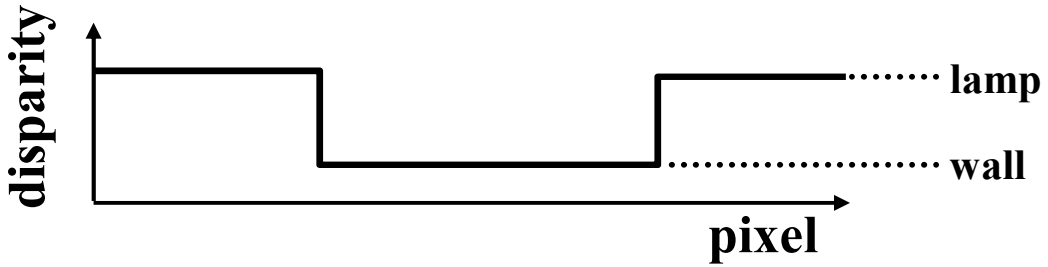


depth discontinuities
(Same layer ?)

Matching Scanlines

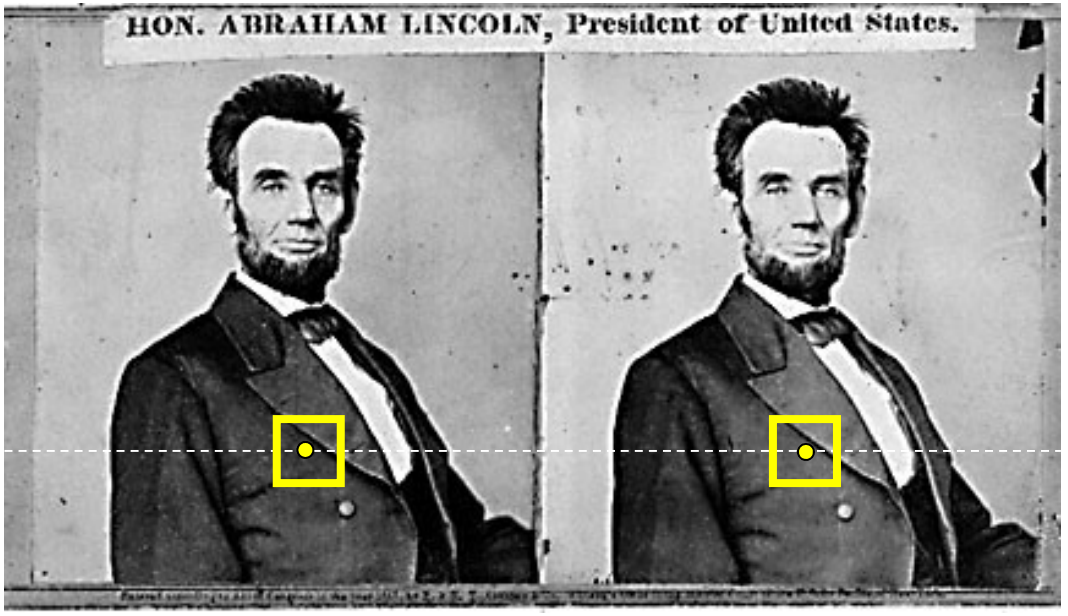






Matching Windows

***See Stereo Basics Page: Left-right image sequence taken at Visionics**



For each epipolar line

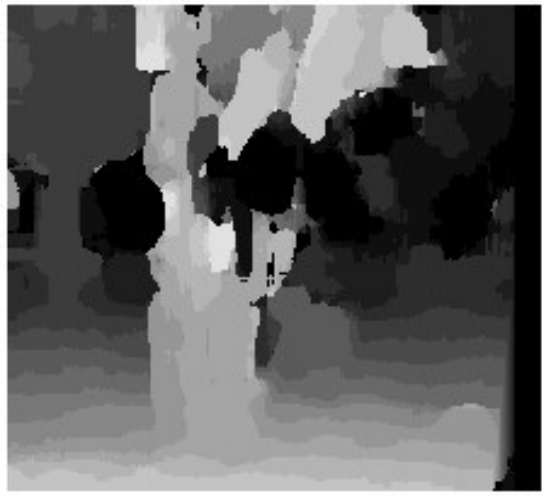
For each pixel in the left image

- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost

Improvement: match ***windows***

- This should look familar...
- Can use Lucus-Kanade or discrete search (latter more common)

Window Size



W = 3 W = 20

- **Effect of window size**
  - Smaller window, depth image:
    - » more details
    - » more noise
  - Larger window, depth image:
    - » less noise
    - » less detail

Better results with *adaptive window*

- *Example:*
  Result more texture, small window size.
  Result less texture, large window size.
- T. Kanade and M. Okutomi, *"A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment*," Proc. International Conference on Robotics and Automation, 1991.
- D. Scharstein and R. Szeliski. "Stereo Matching with Nonlinear Diffusion," International Journal of Computer Vision, 28(2):155-174, July 1998

Match Metrics

- **Some commonly used match metrics**

1) SAD - Sum Absolute Difference

$$\min \quad \psi(x,y) = \sum_{i \in w} | x_i - y_i |$$

2) SSD - Sum of Squared differences

$$\min \quad \psi(x,y) = \sum_{i \in w} (x_i - y_i)^2$$

3) Cross Correlation

$$\max \quad \psi(x,y) = \sum_{i \in w} (x_i * y_i)$$

4) Normalized Cross Correlation

$$\max \quad \psi(x,y) = \frac{1}{(\sigma_x \sigma_y)} \sum_{i \in w} (x_i - \bar{x})(y_i - \bar{y})$$

$$\min \frac{\sum (I_1 - I_2)^2}{\sqrt{\left(\sum I_1^2 \sum I_2^2\right)}}$$

$$\max \frac{\sum I_1 * I_2}{\sqrt{\left(\sum I_1^2 \sum I_2^2\right)}}$$

$$\min \frac{\sum \left((I_1 - \bar{I}_1) - (I_2 - \bar{I}_2)\right)^2}{\sqrt{\sum (I_1 - \bar{I}_1)^2 \sum (I_2 - \bar{I}_2)^2}}$$

$$\max \frac{\sum \left((I_1 - \bar{I}_1)(I_2 - \bar{I}_2)\right)}{\sqrt{\sum (I_1 - \bar{I}_1)^2 \sum (I_2 - \bar{I}_2)^2}}$$

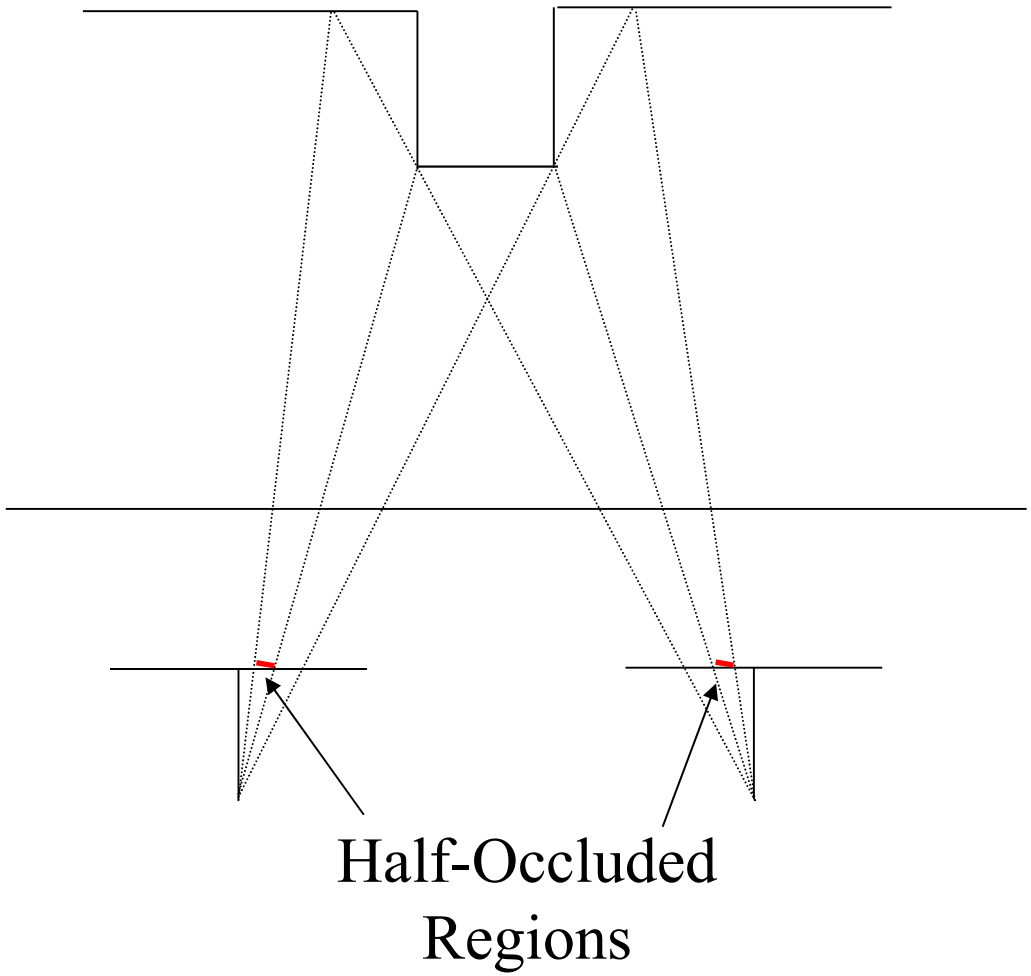
- **Consider time complexity**

Correlation Techniques:

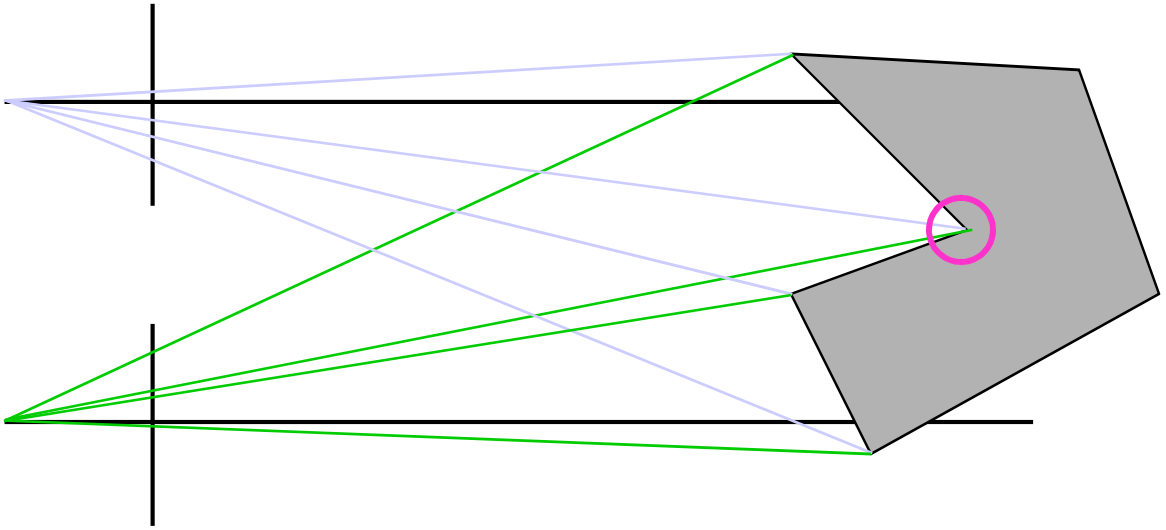
Half Occluded Regions

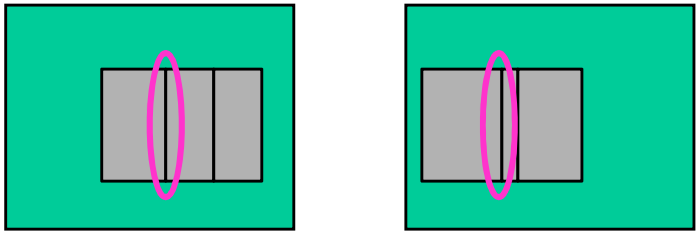
- In certain situations, there may be points in the scene that are visible from one of the two cameras but not the other. These are referred to as half-occluded regions. Due to
  1. Two cameras have different views
  2. Occlusion
  3. Missing points due to feature detection techniques
- These regions are frequently associated with depth discontinuities in the scene
- Half occluded regions pose a challenge to most simple stereo correspondence algorithms
- Consequently many stereo algorithms have problems at depth discontinuities
- The human visual, by contrast, seems to be sensitive to half-occluded regions (depth discontinuities ?) and uses them to detect surface boundaries

- **Reducing baseline B (narrow-angle-stereo) can alleviate the correspondence and occlusion problem, but it may lead to less accurate depth estimate.**

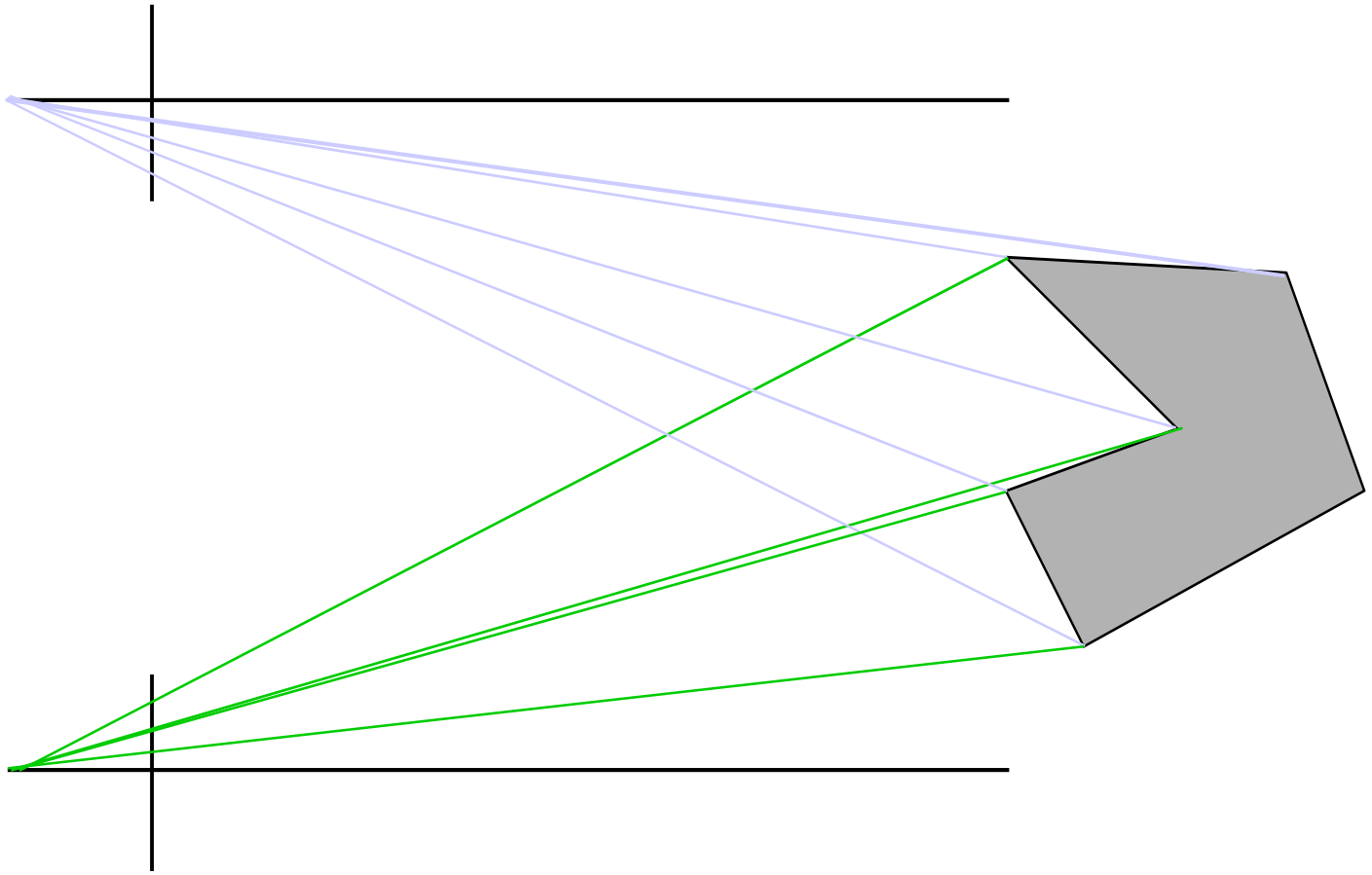


Different Baselines

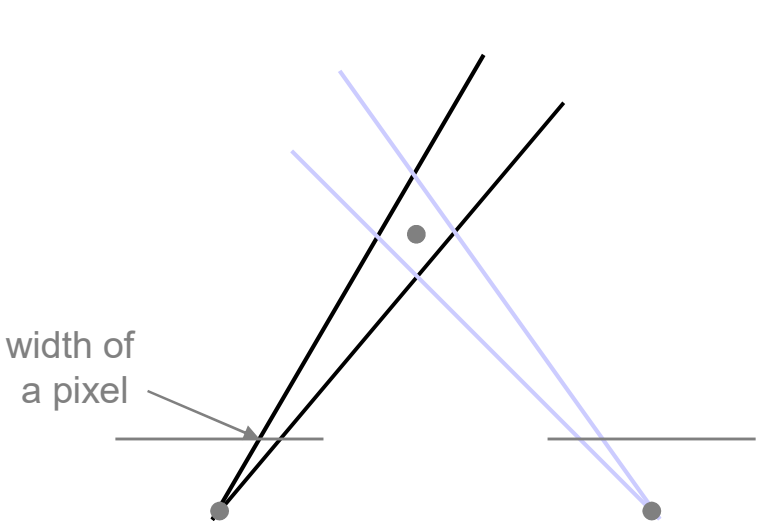




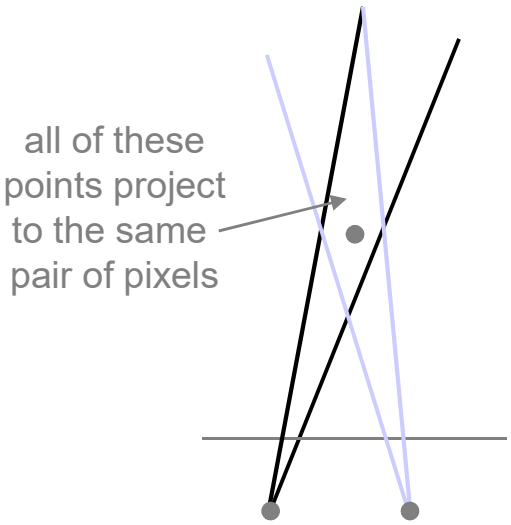
Wider Baseline:



Choosing the Baseline



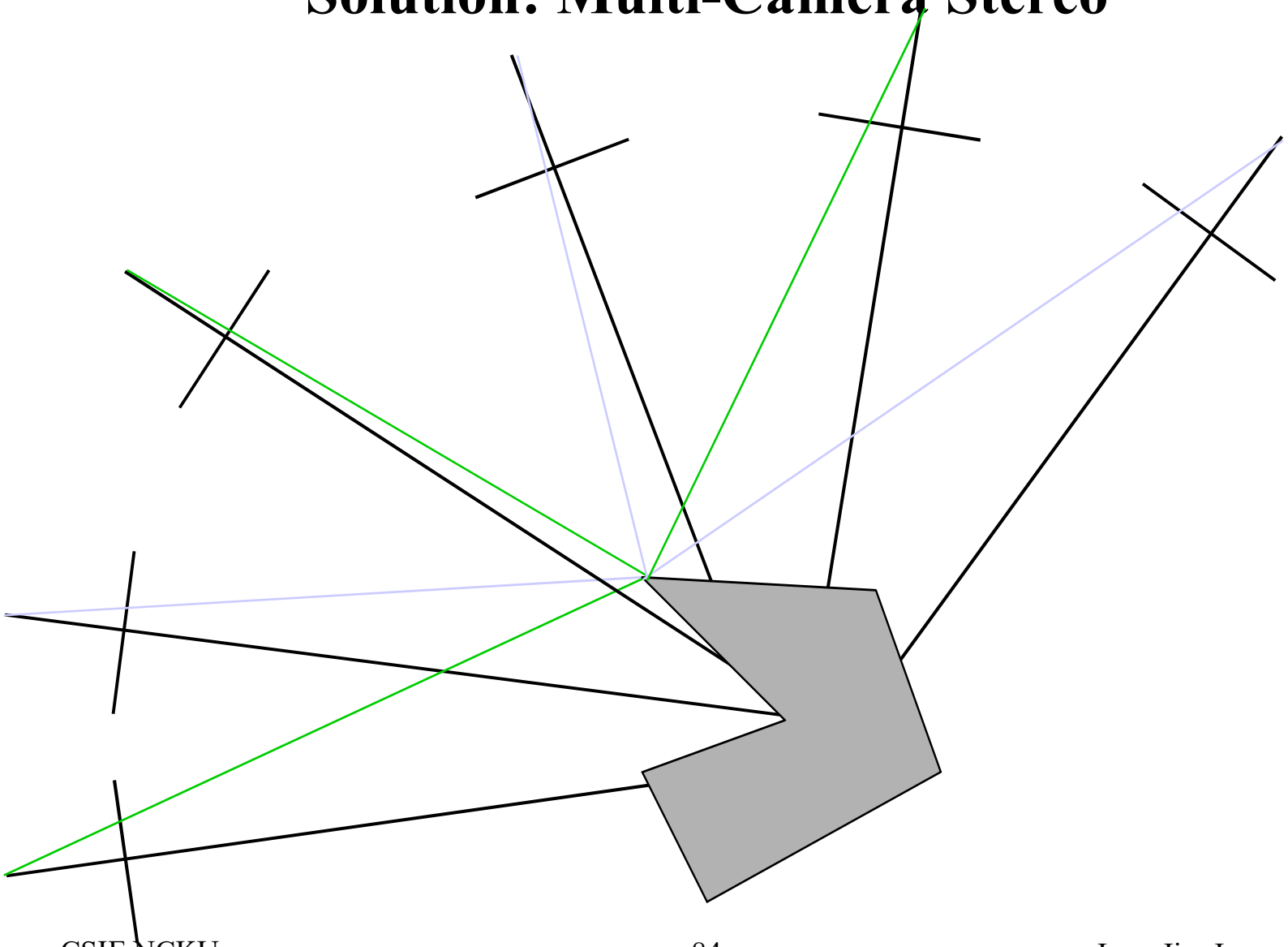
Large Baseline



Small Baseline

- **What's the optimal baseline?**
  - Too small: large depth error
  - Too large: difficult search problem

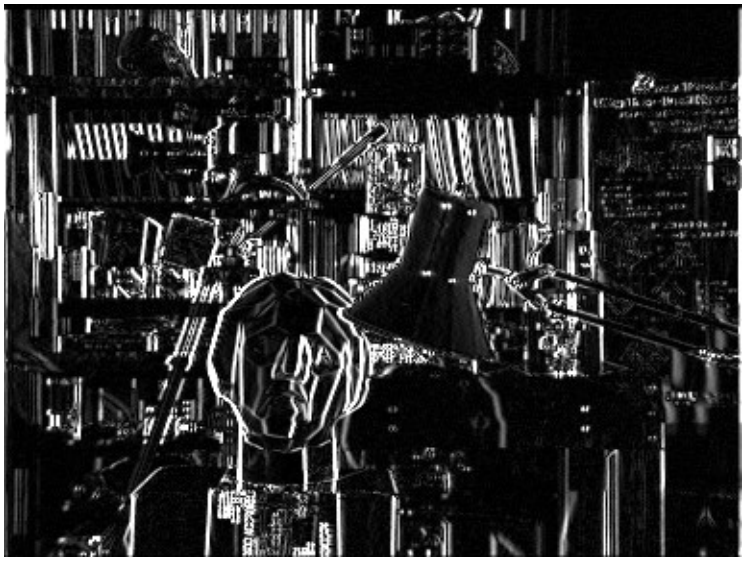
Solution: Multi-Camera Stereo



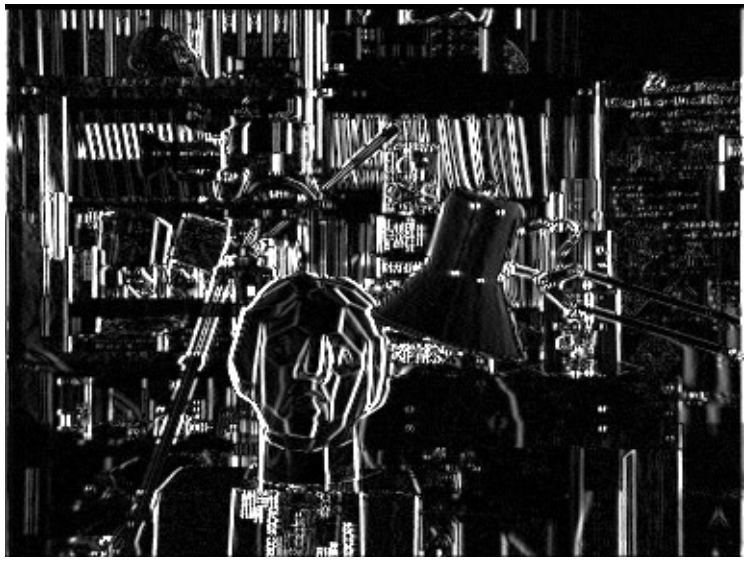
Left-Right Consistency Constraint

- **One approach to detect half-occluded regions is**
  - **to run the stereo procedure in both directions.**
  - **From left to right and right to left, and then to check that correspondences are consistent in both solutions.**

Figural Continuity Constraint

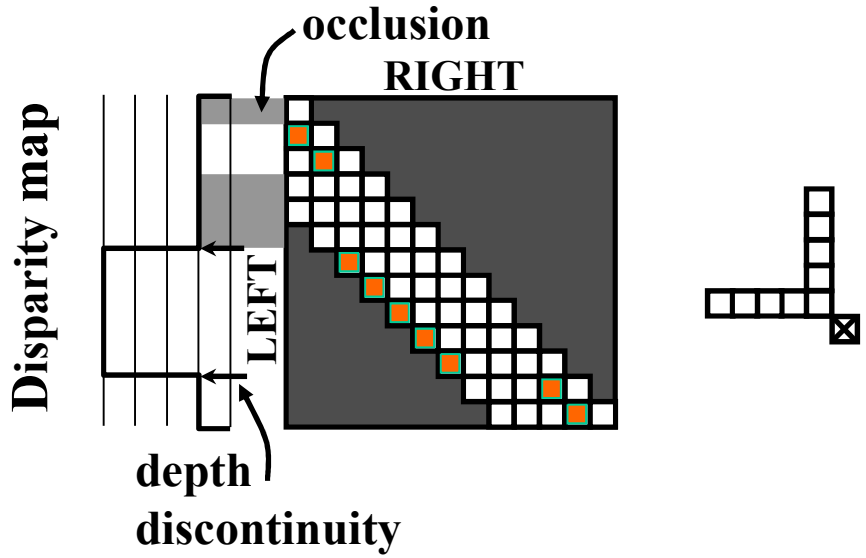


right



left

Dynamic Programming

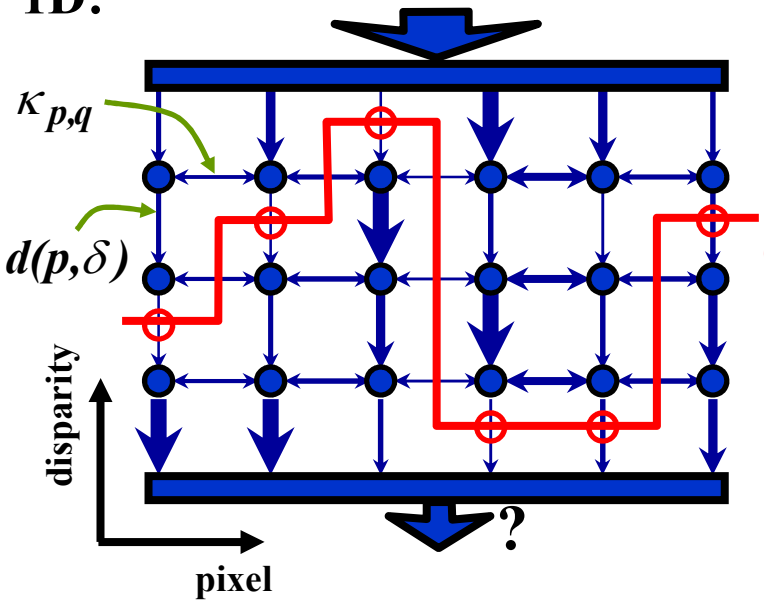


Minimizing a 2D Cost Function:

$$\underset{\delta}{\text{Minimize:}}\ E_{data}+E_{smoothness}=\sum_{p} d(p,\delta)+\sum_{\{p,q\}\in N}\kappa_{p,q}u(l_{p,q})$$

GLOBAL

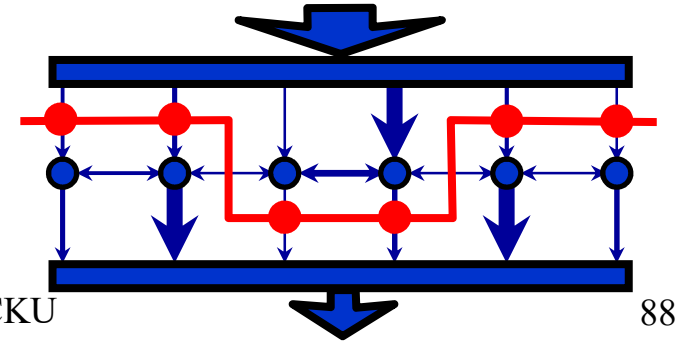
1D:



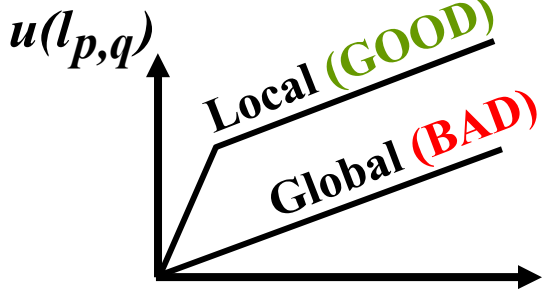
minimum cut = disparity surface

solves $u(l_{p,q})=\kappa_{p,q}l_{p,q}$

LOCAL



Discontinuity penalty:

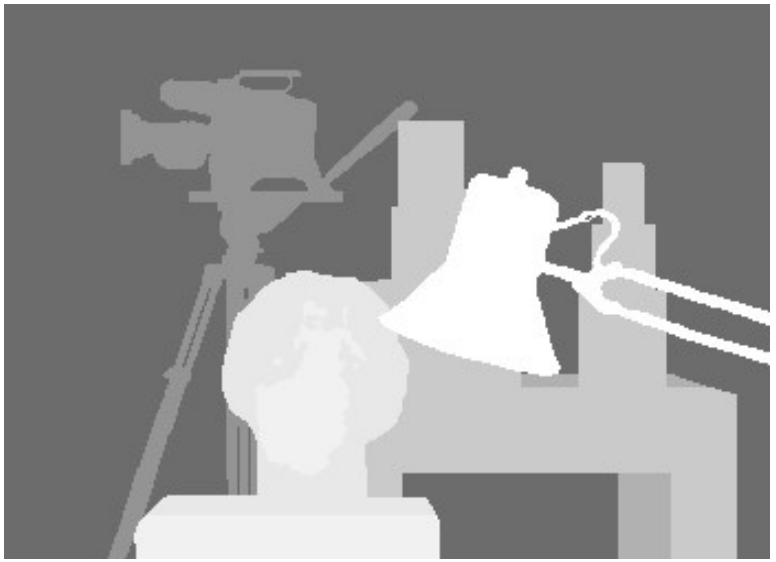


Stereo Examples

Data from University of Tsukuba:

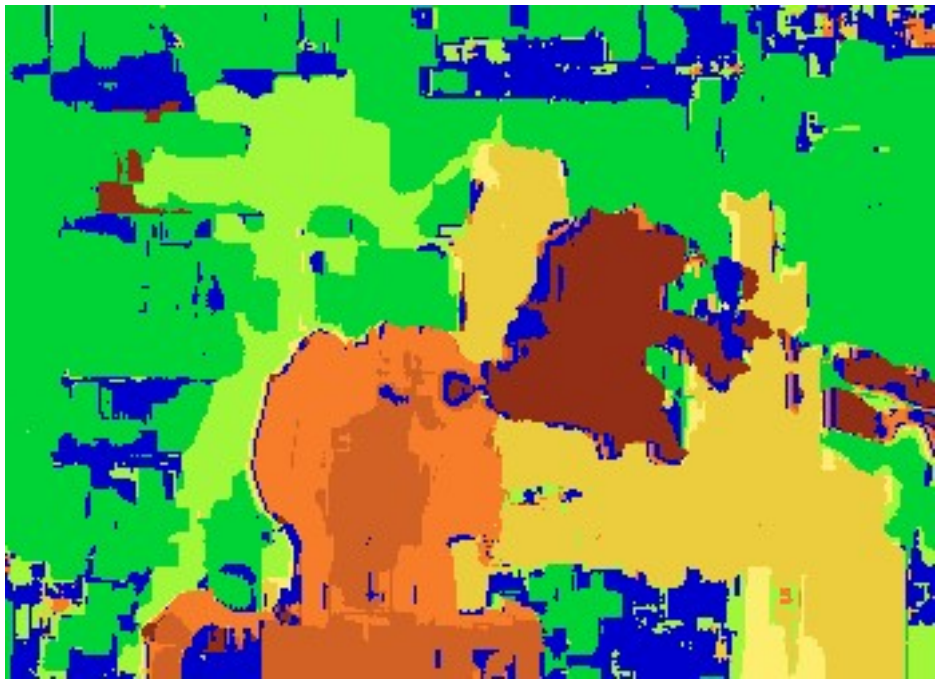
- Similar results on other images without ground truth

Scene



Ground truth

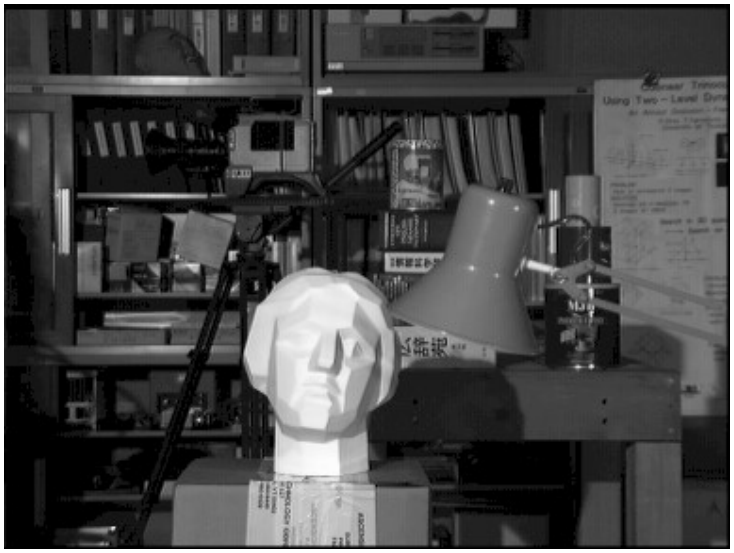
Results with window correlation -

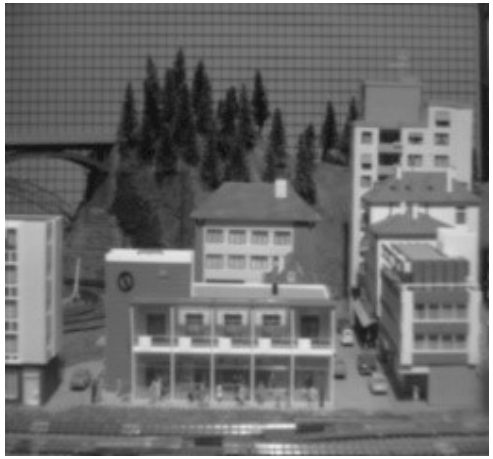


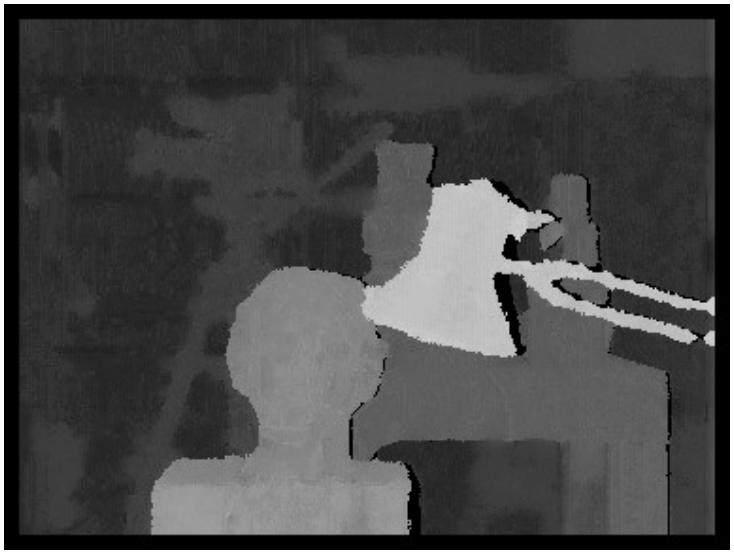
Window-based matching
(best window size)

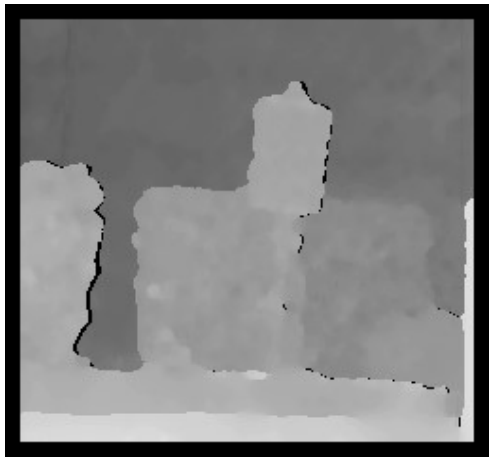
Ground truth

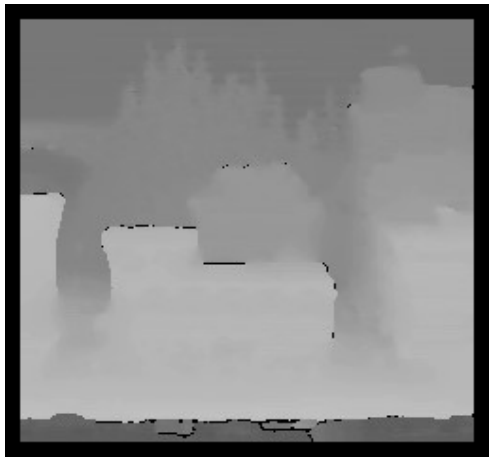
Ground truth



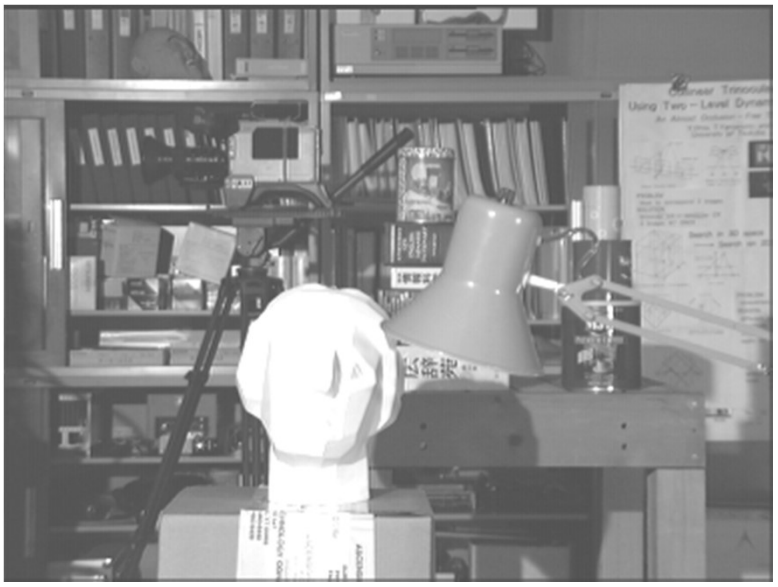








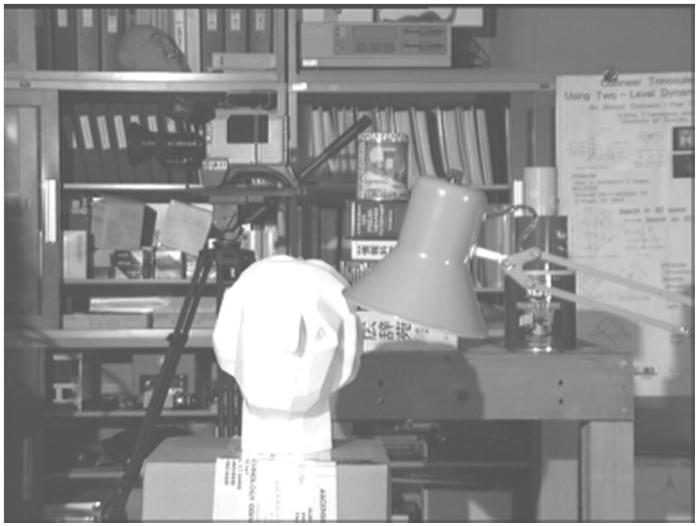
Results: Correlation



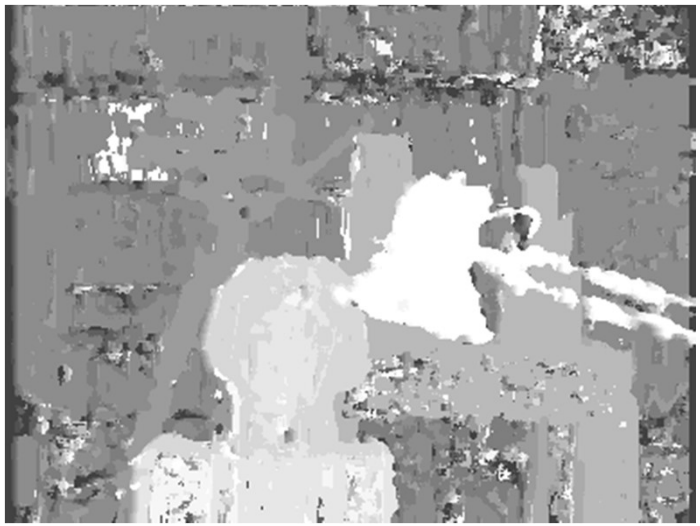
left

disparity map

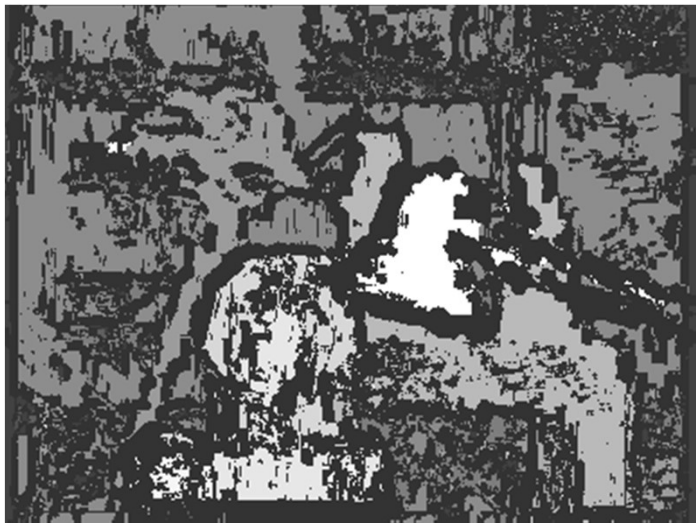
Results: Left-Right Consistency Check



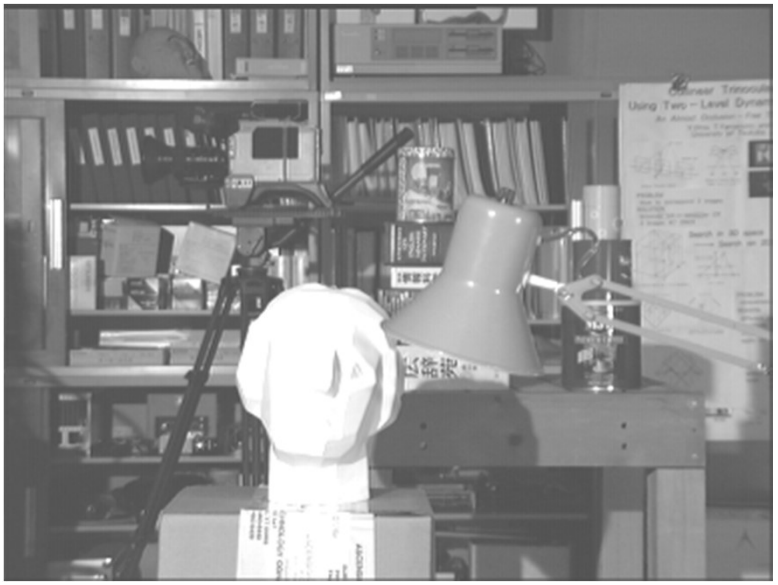
without →



with →



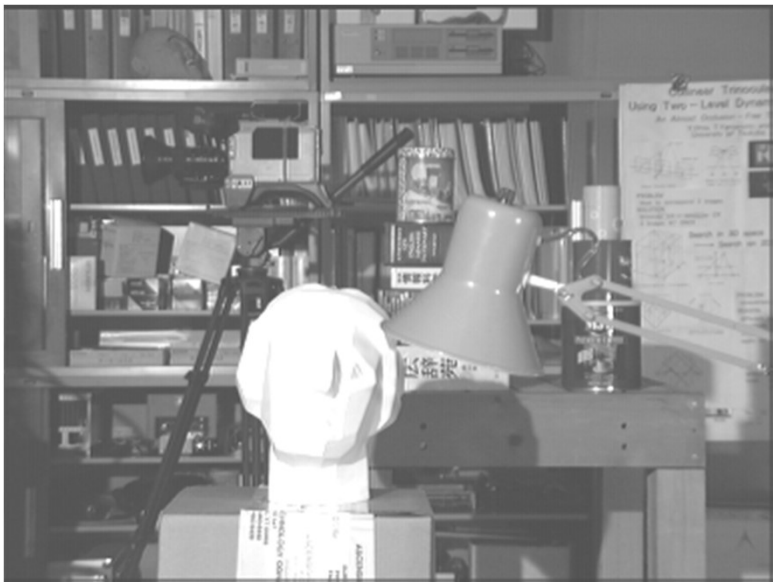
Results: Dynamic Programming



left

disparity map

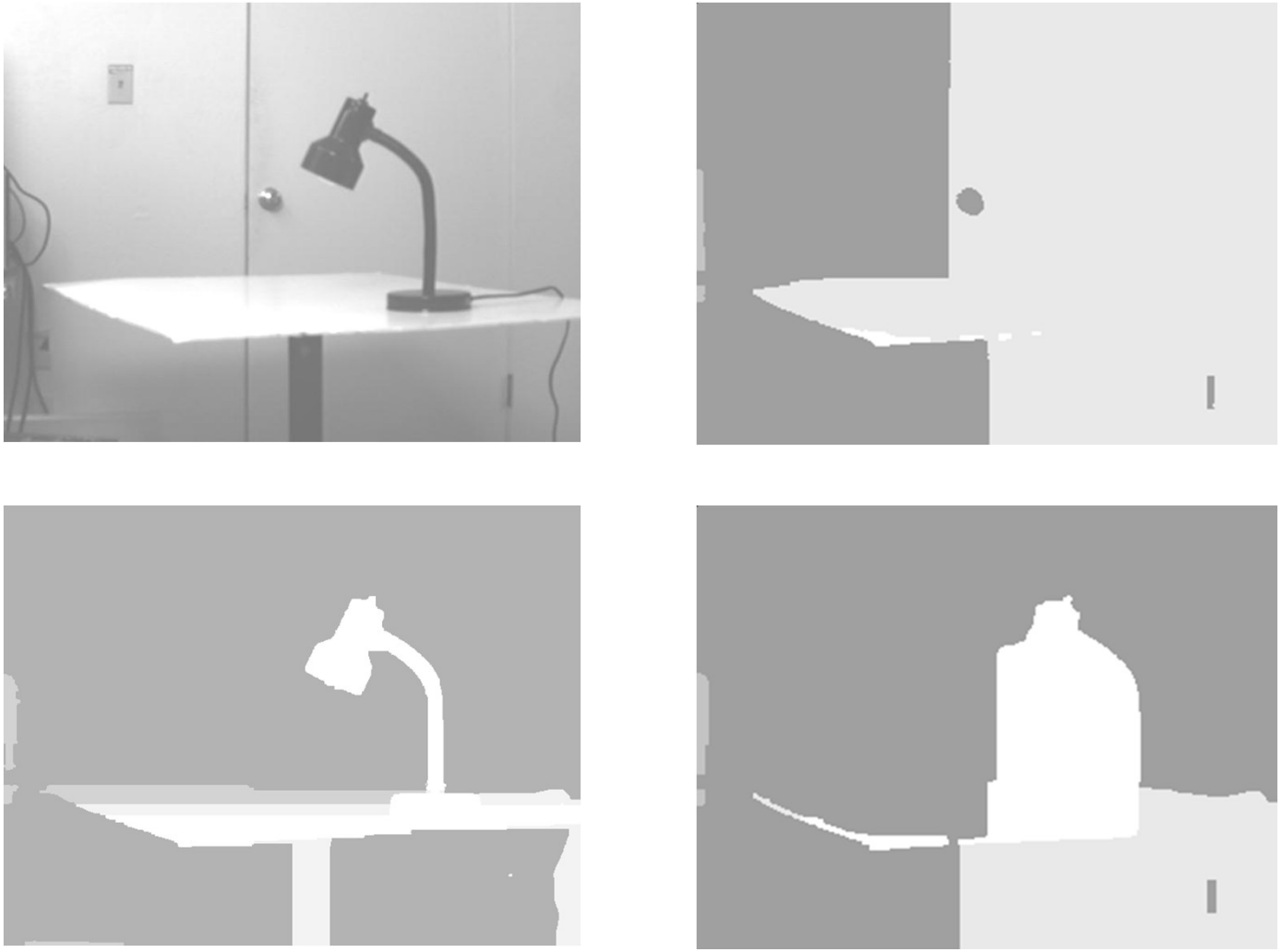
Results: Multiway Cut



left

disparity map

Results: Multiway Cut (Untextured)



Stereo Reconstruction Pipeline

- **Steps**
  1) Calibrate cameras
  2) Rectify images
  3) Compute disparity *d*: Window size, search range
  4) Estimate depth Z=(f*B)/d

- **What will cause errors?**
  1) Camera calibration errors
  2) Poor image resolution
  3) Occlusions
  4) Violations of brightness constancy (specular reflections)
  5) Large motions
  6) Low-contrast image regions

Real Stereo Vision System

1. **Preprocessing: LOG transform**
2. **Matching/Correlation**
3. **Calibration**
4. **Variable disparity search**
5. **Post-filtering with an interest operator**
6. **Left-Right check**
7. **Depth interpolation**

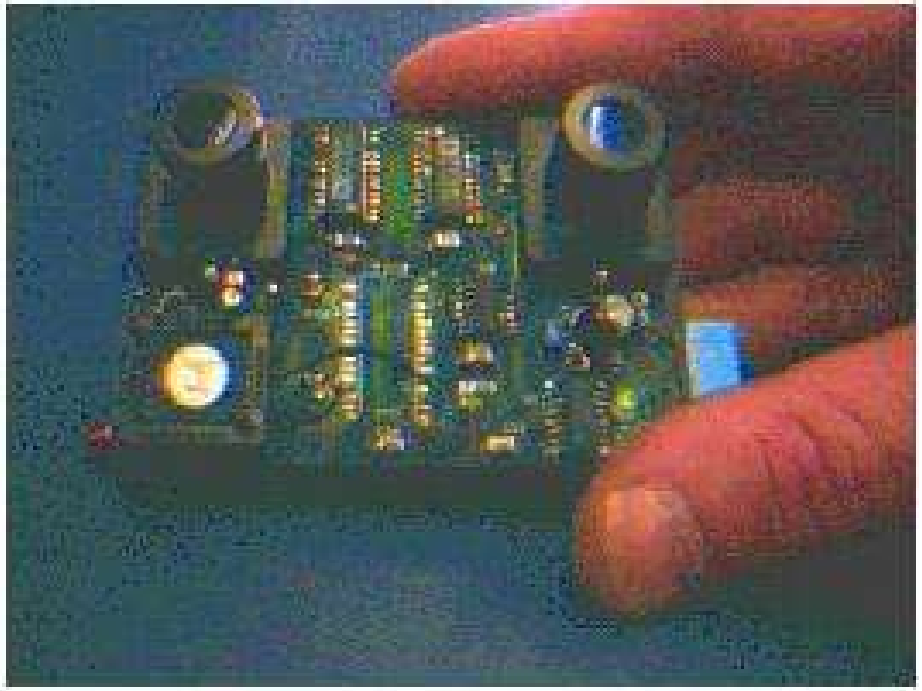


Figure 7 The SRI Small Vision Module

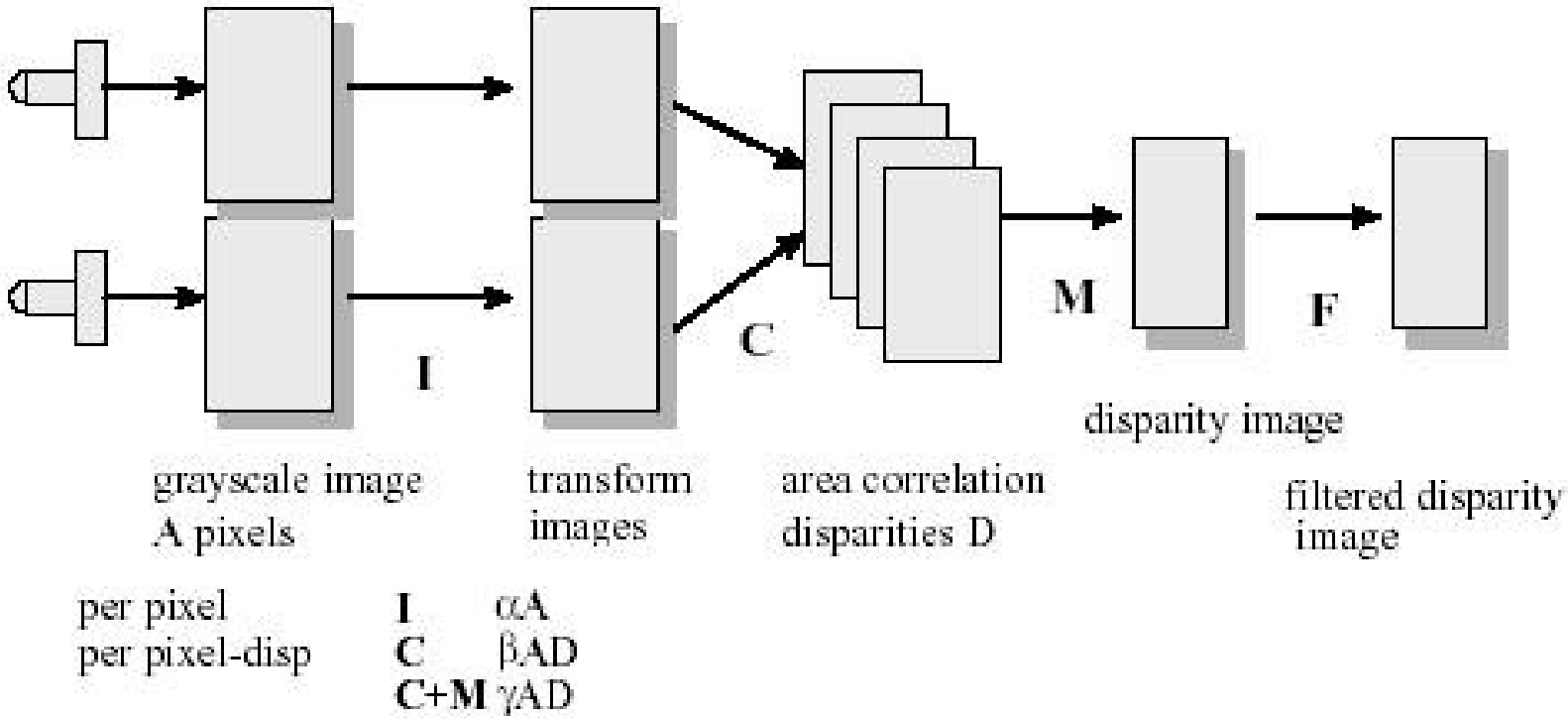
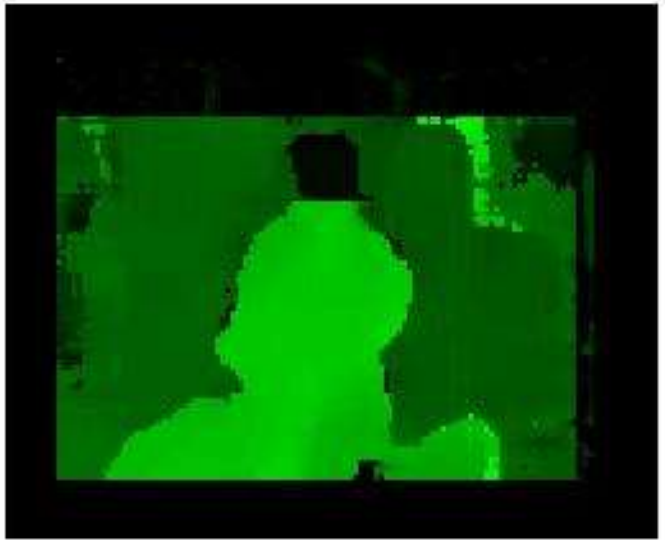
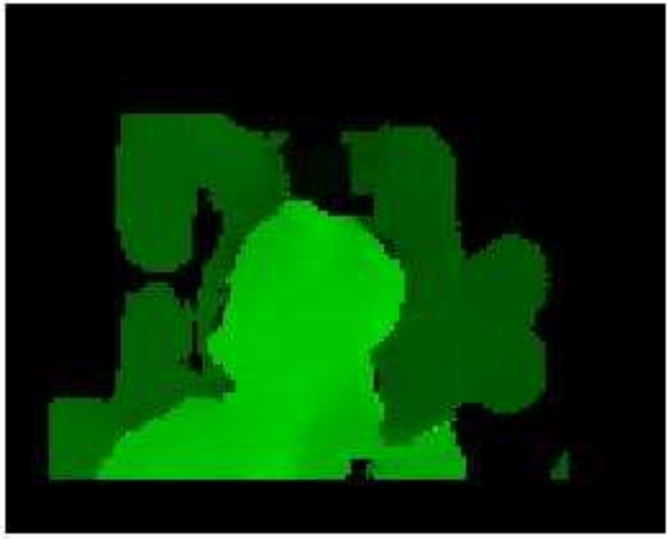


Figure 3 Area correlation algorithm diagram. *A* is the area of an image, and *D* is the number of disparities searched over. Image warping and post-filtering costs are not considered here. The cost of the algorithm is divided into per pixel and per pixel-disparity contributions.

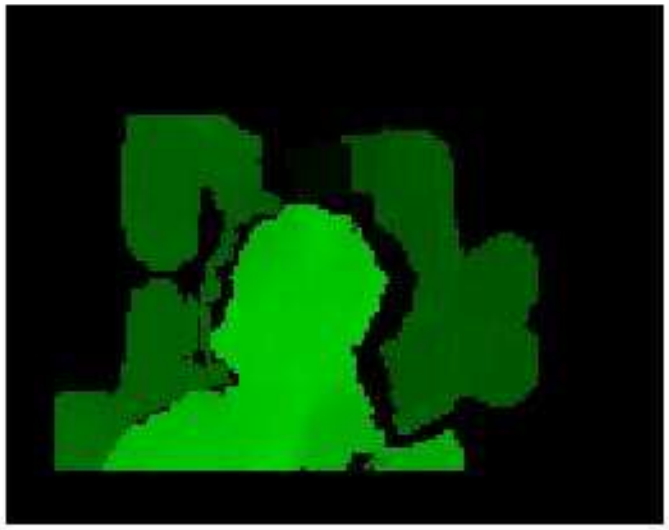
(a) Input grayscale image, one of a stereo pair



(b) Disparity image from area correlation



(c) Texture filter applied



(d) Left/right and texture filter applied

Figure 2 A grayscale input image and the disparity images from the SRI algorithm.

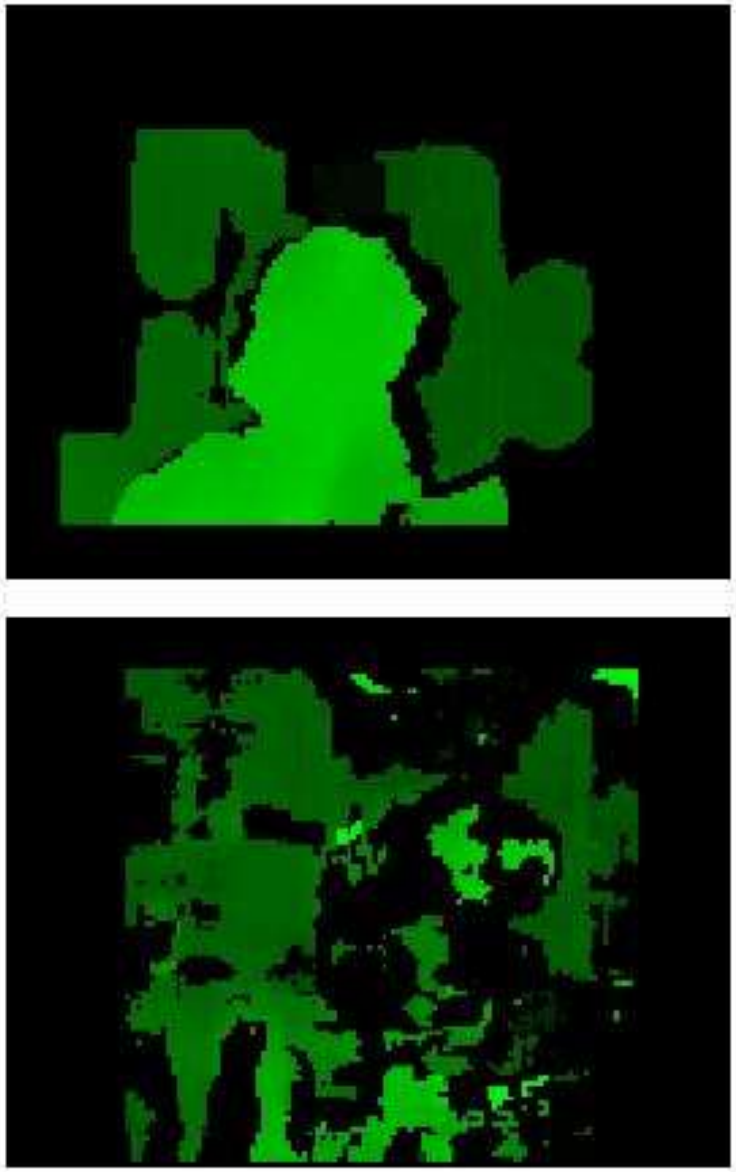


Figure 4 Calibrated and uncalibrated disparity images. The stereo input to the bottom image was offset 2 vertical pixels from the top.

LR criterion

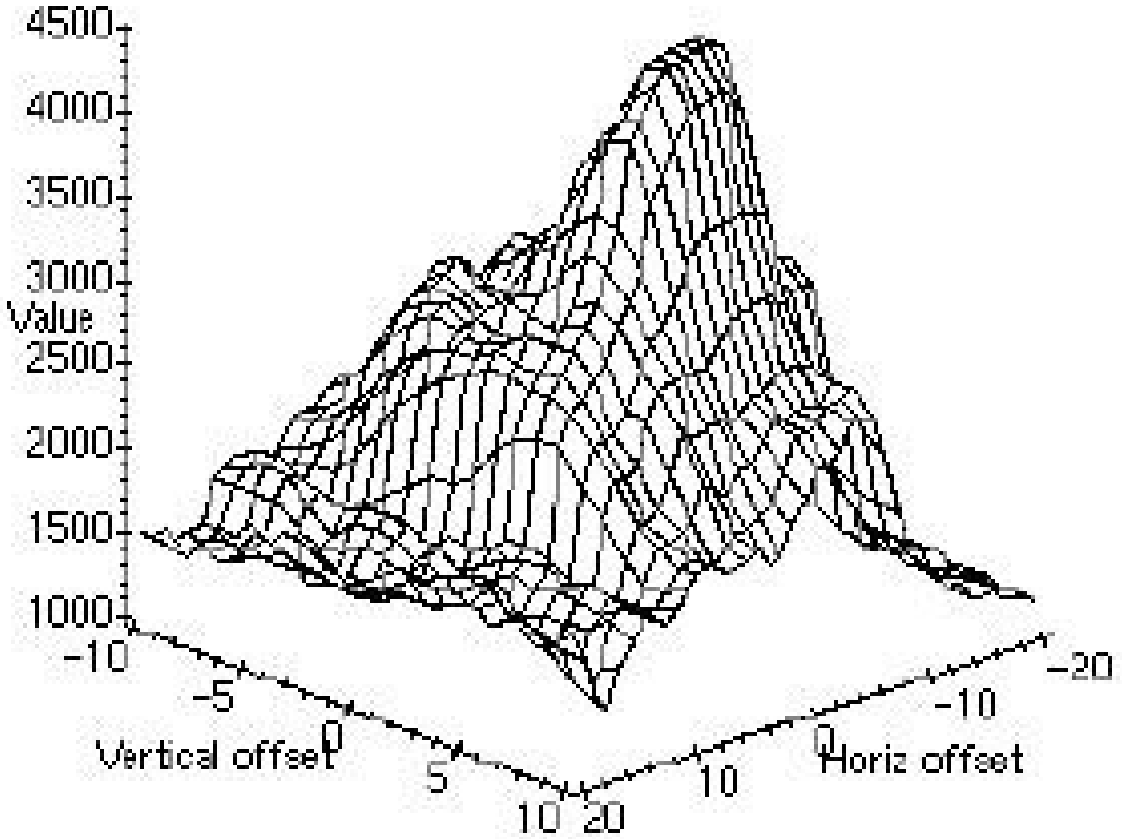


Figure 5 Number of left/right matches as a function of horizontal and vertical offset.

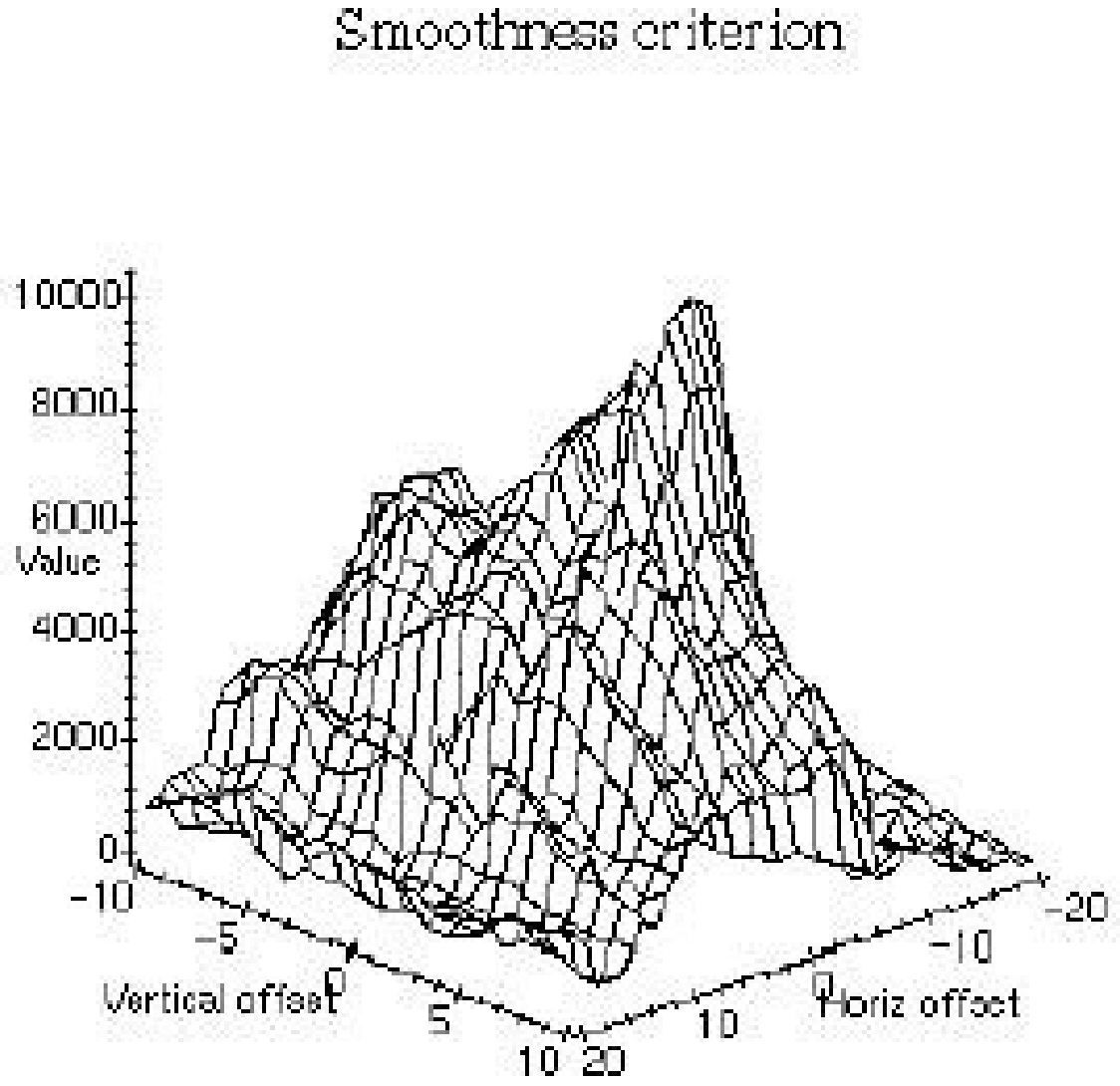


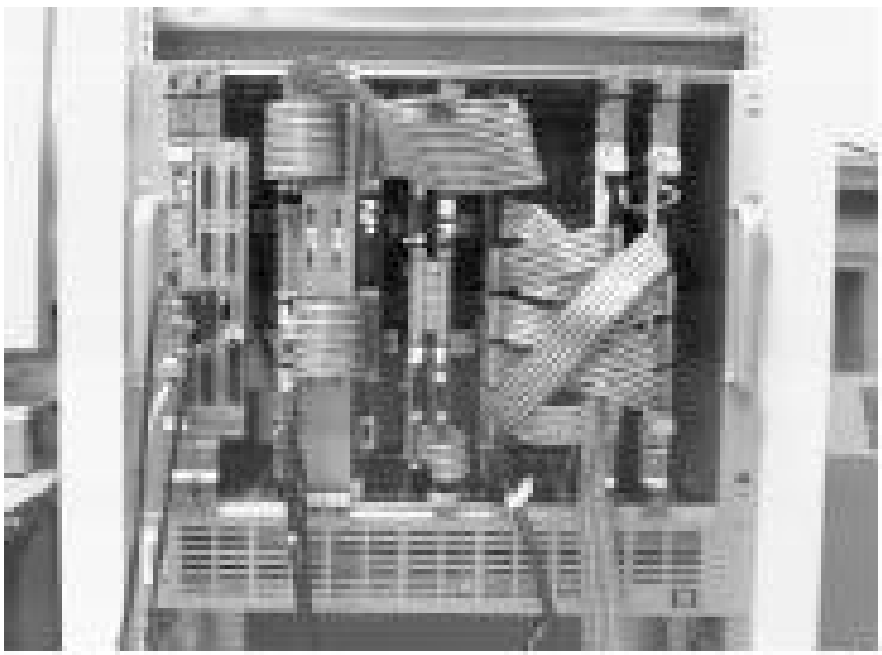
Figure 6 Smoothness of disparity image as a function of horizontal and vertical offset.

Multi-baseline Stereo

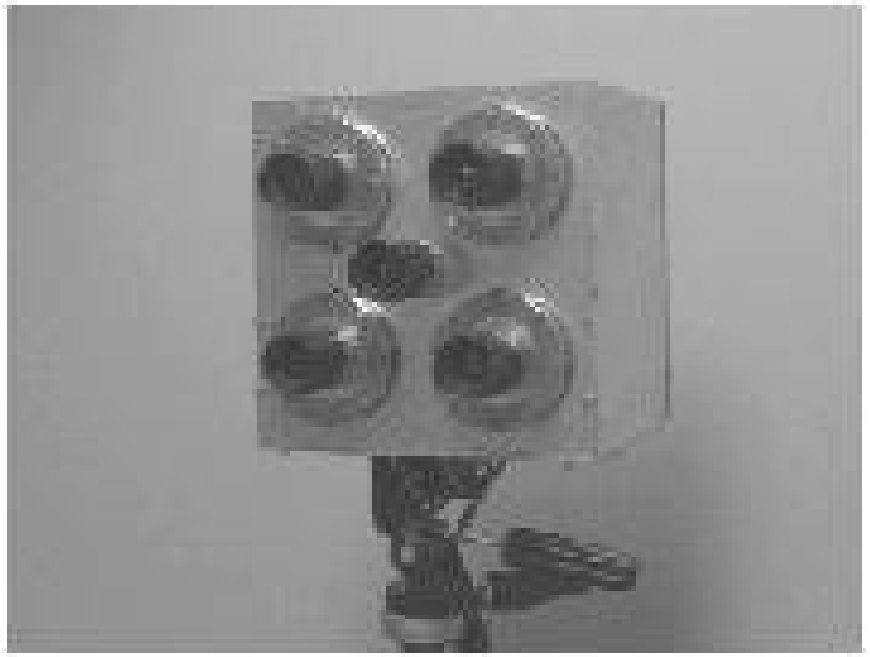
Point Grey Research

A Multiple-Baseline Stereo (CMU-Takeo)

- **CMU Video-Rate Stereo Machine and Its Performance:**

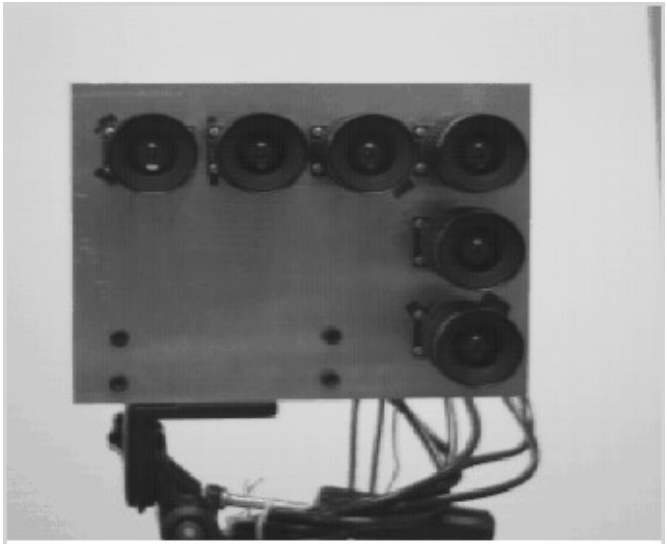


(a) Processor



(b) Five-eye camera head

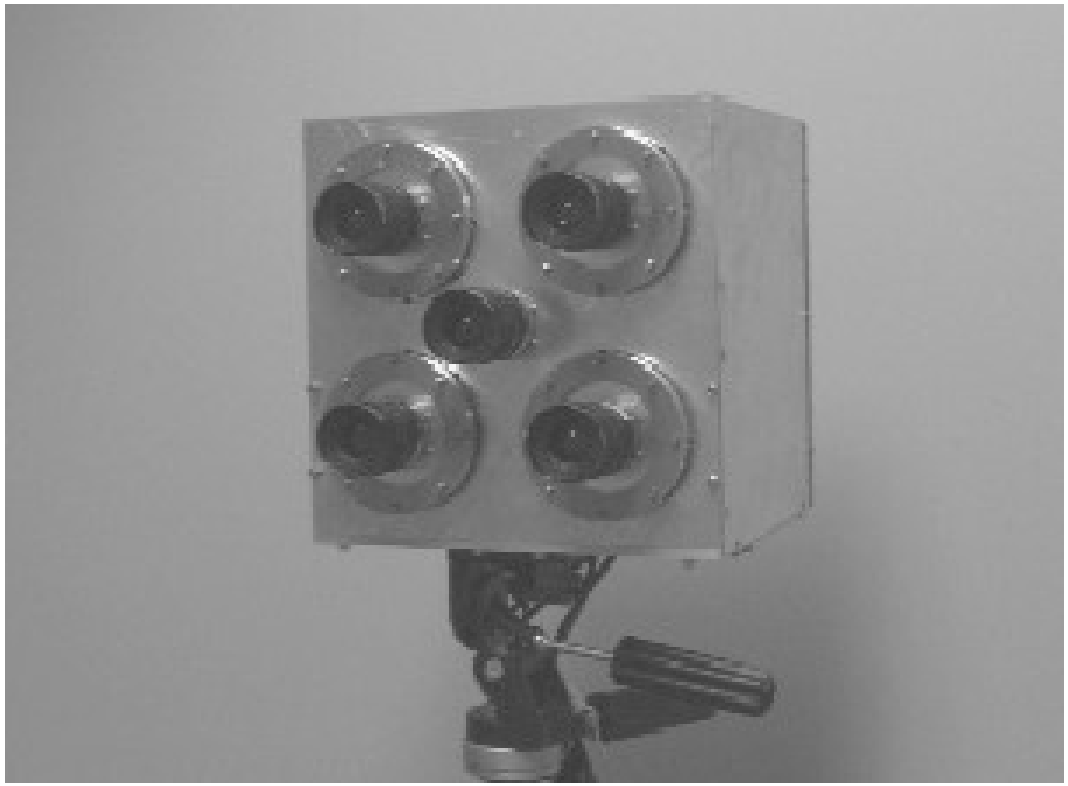
Figure 1: The CMU video-rate stereo machine



(a)

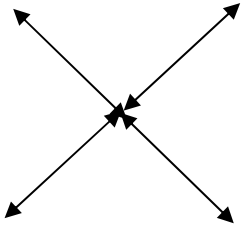
(b)

Figure 7: The CMU Video-Rate Stereo Machine Prototype System: (a) camera head; (b) processor boards



Multi-baseline stereo

Old Vs. New



(a) intensity image

(b) corresponding disparity map

Figure 2: An example scene and its range image

Table 1: Performance of CMU Stereo Machine

Number of Cameras:	**2 to 6**
Processing time/pixel:	**33 ns x (disparity range + 2)**
Frame rate:	**up to 30 frames/sec**
Depth image size:	**up to 256 x 240**
Disparity search range:	**up to 60 pixels**

❑Multi-Baseline Stereo Algorithm:

Theory -

$$\frac{d}{B} = F \cdot \frac{1}{z} = \xi$$

$$SSD_k(i,j,\xi) = \sum_{(s,t)\in W(i,j)} SD_k(s,t,\xi)$$

$$= \sum_{(s,t)\in W(i,j)} \left(f_k\left(s + c_1 \cdot (B_k \cdot \xi), t + c_2 \cdot (B_k \cdot \xi)\right) - f_0(s,t) \right)^2$$

$$SSSD(i,j,\xi) = \sum_{k=1}^{n} SSD_K(i,j,\xi) = \sum_{k=1}^{n} \left(\sum_{(s,t)\in w(i,j)} SD_k(s,t,\xi) \right)$$

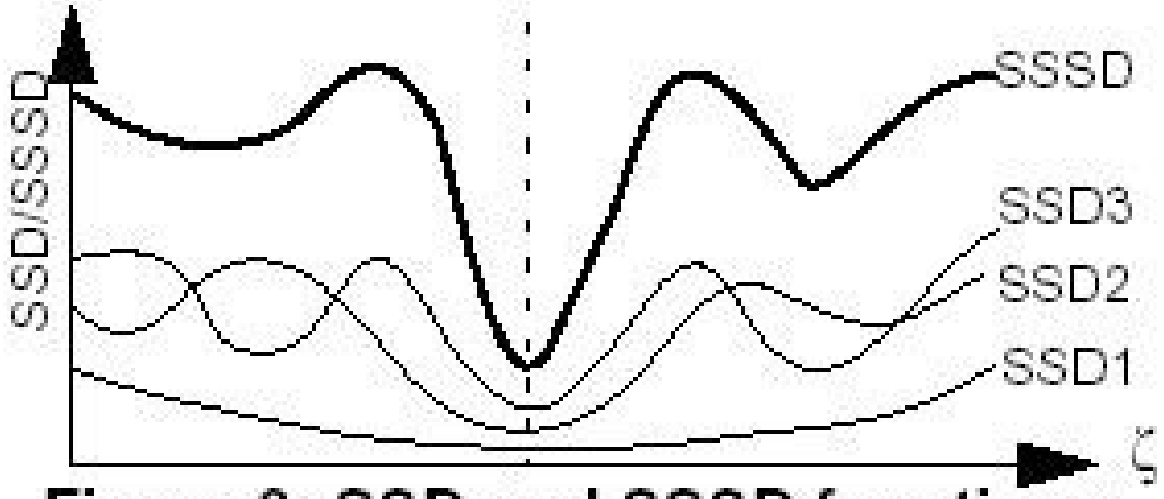


Figure 3: SSD and SSSD functions

Summary of the Algorithm -

The total amount of computation per second required for the SSD calculation:

$$N^2 \times W^2 \times D \times (C-1) \times P \times F$$

N^2:	The image size
W^2:	The widow size
D:	The disparity range (i.e., search range)
C:	The number of cameras
P:	The number of operation per one SD calculation
F:	The number of frames per second

If N=256, W=11, D=30, C=6, and F=30, then the total computation would be 465 giga-operations

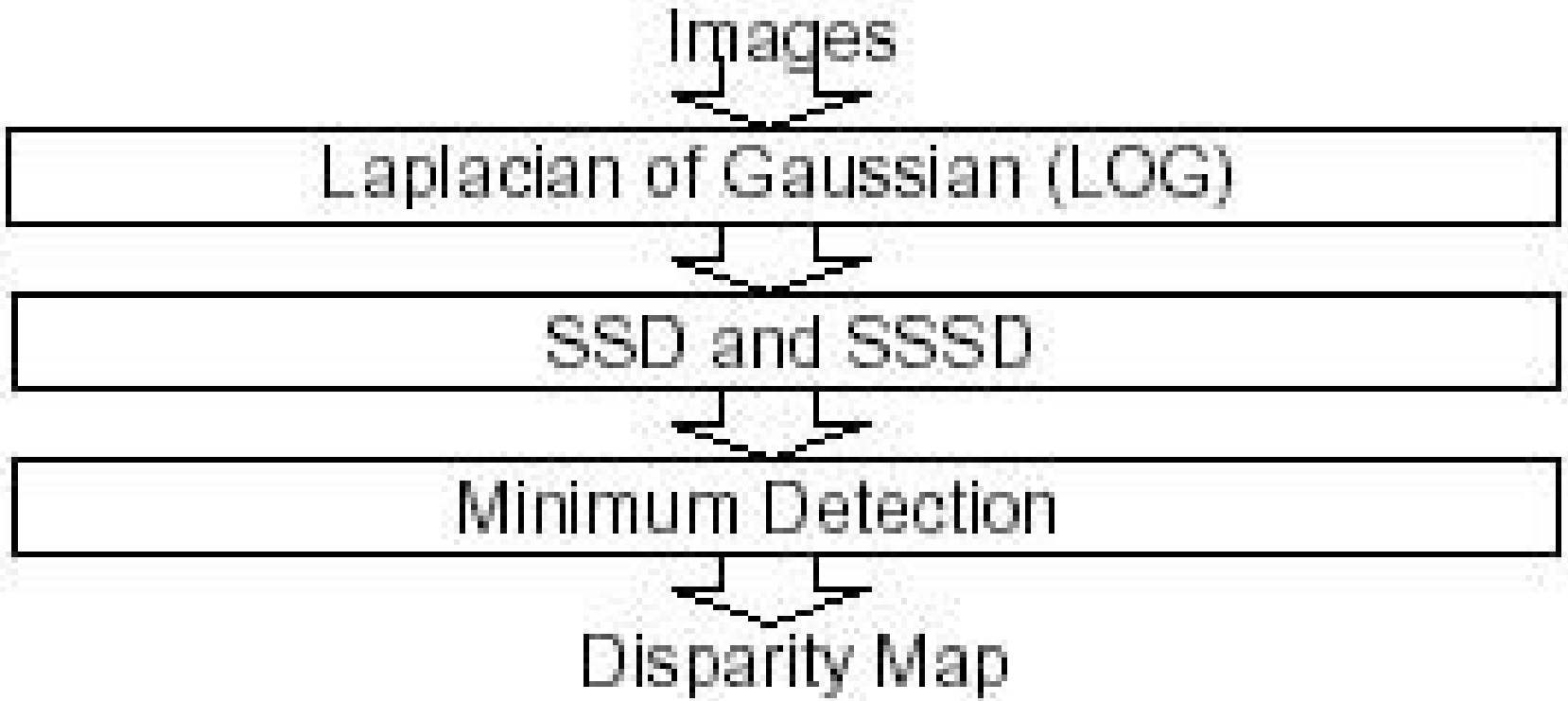


Figure 4: Outline of stereo method

❑Design of a

Video-Rate Stereo Machine:

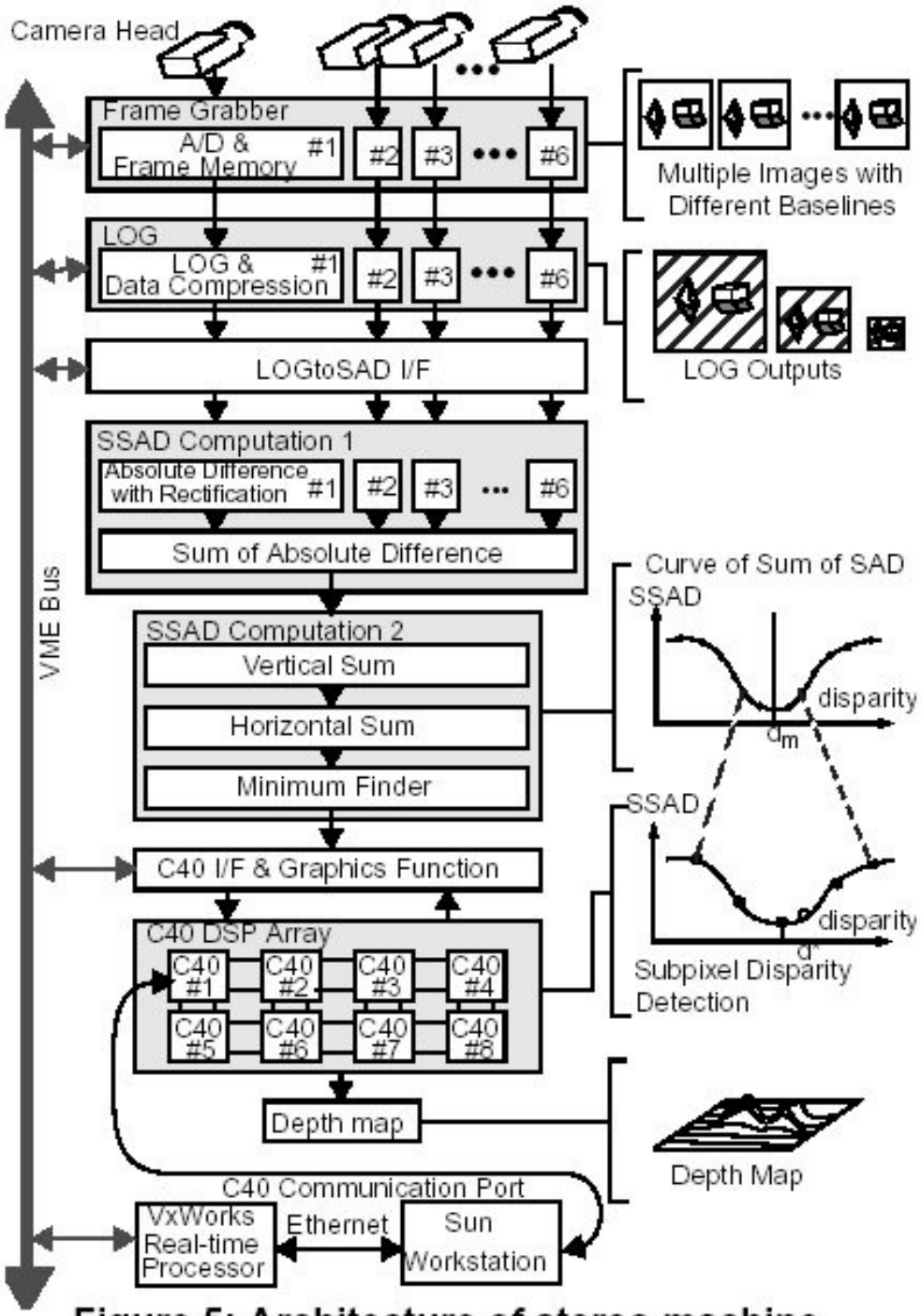


Figure 5: Architecture of stereo machine

LOG Subsystem -

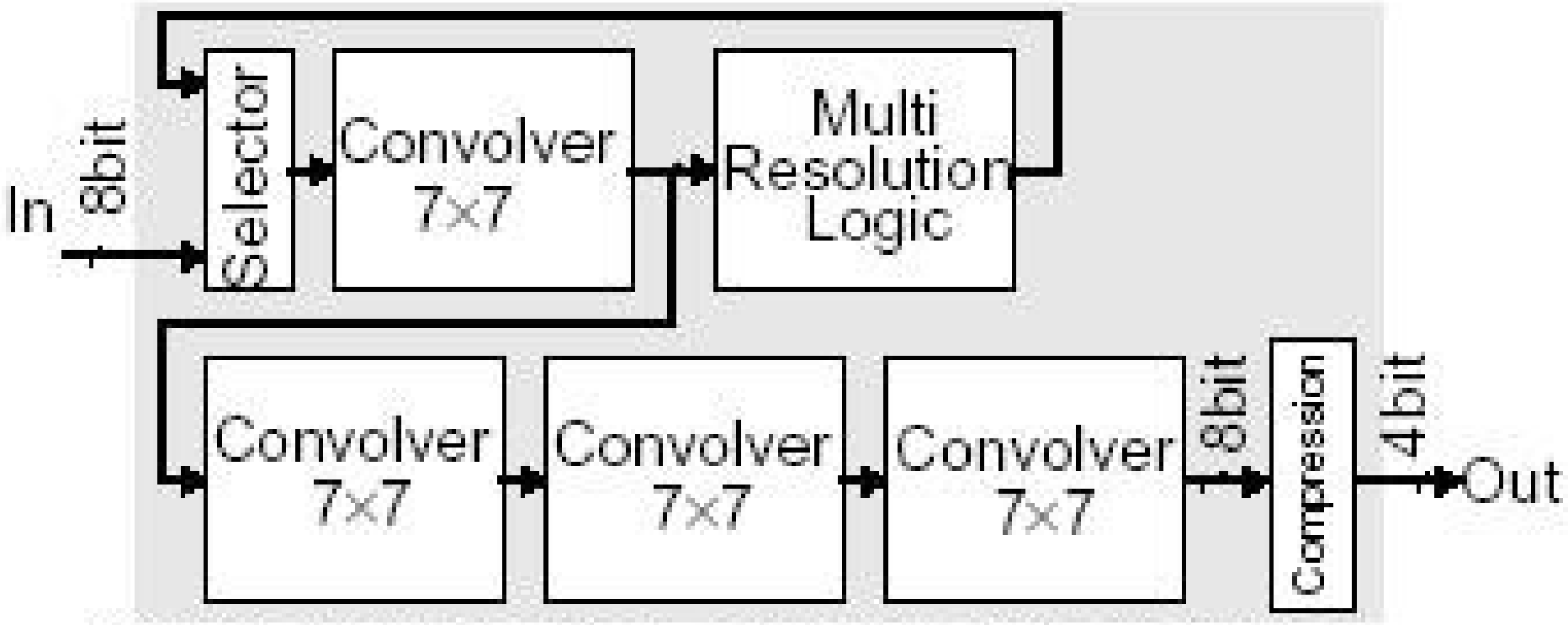


Figure 6: Function of LOG subsystem

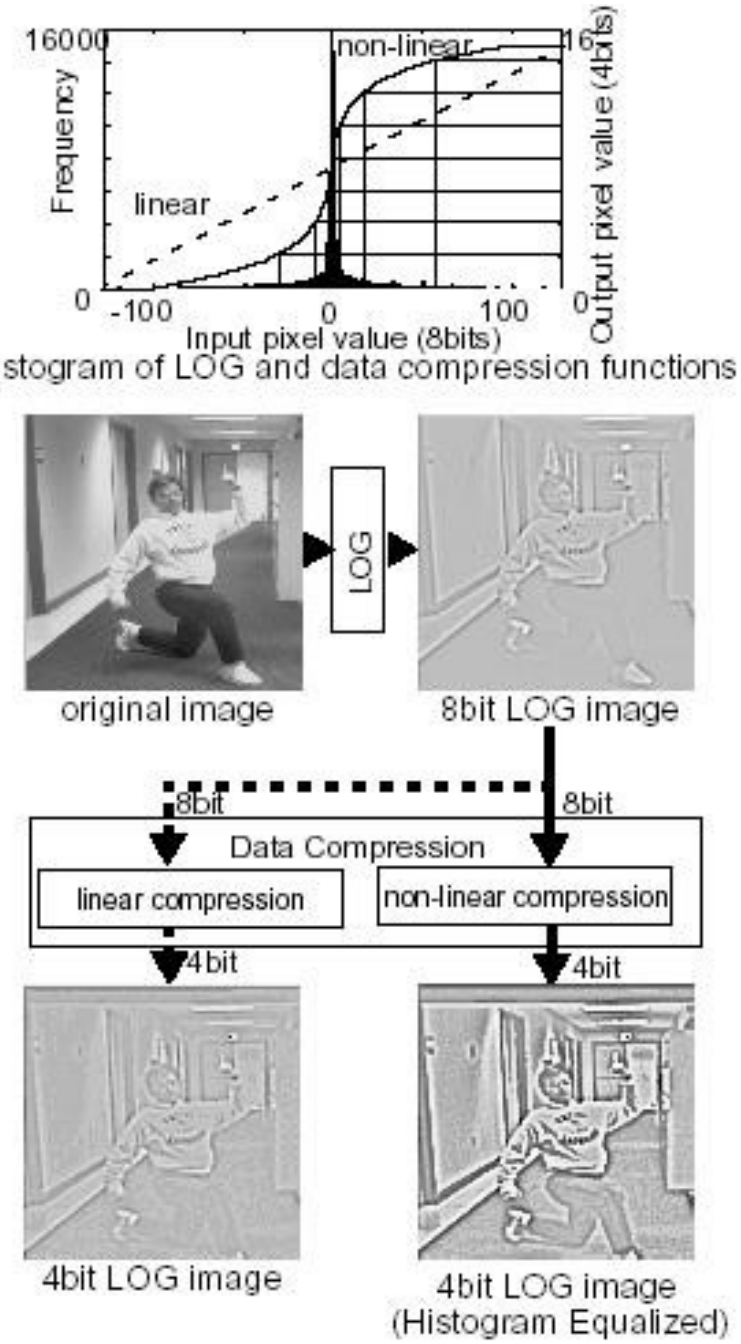


Figure 7: 8bit to 4bit Data Compression of LOG image

SSAD Subsystem –

1. Rectification of Images

$$AD_k(s,t,\xi) = f_k(I_k(s,t,\xi), J_k(s,t,\xi)) - f_0(I_0(s,t), J_0(s,t))$$

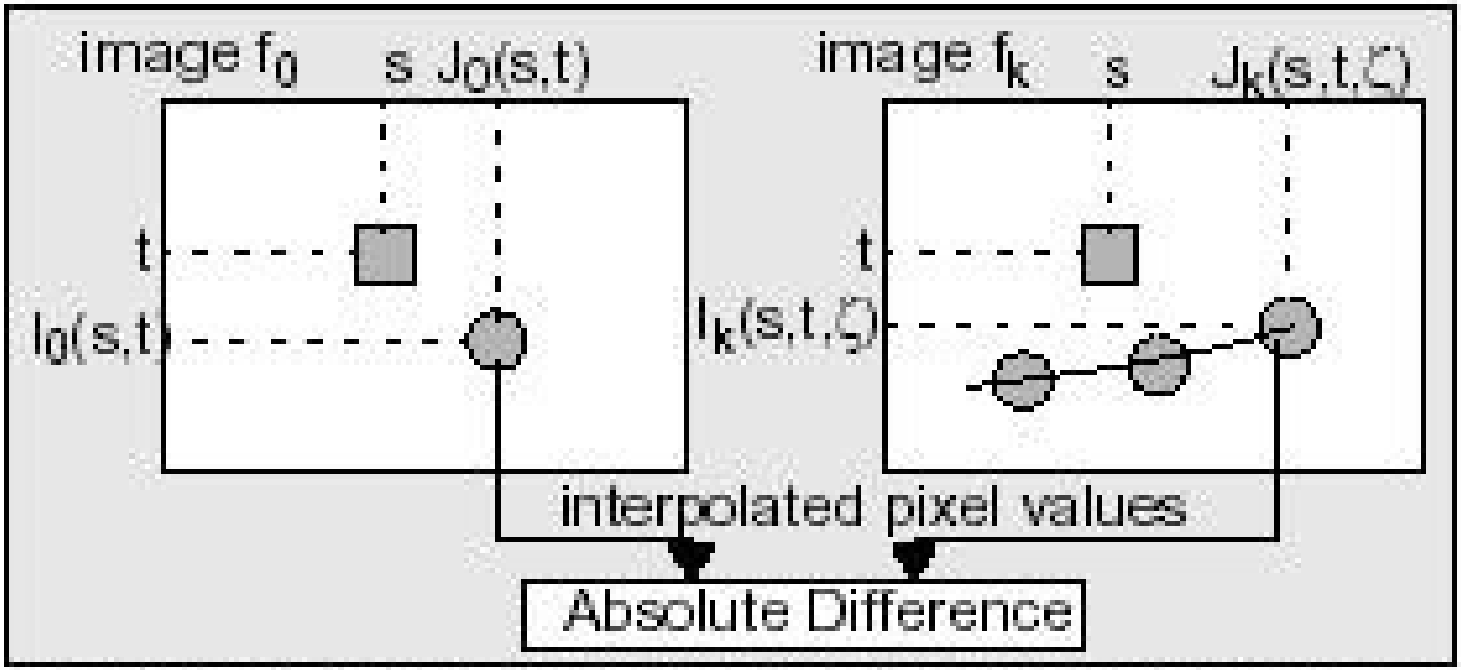


Figure 8: Calculation of Absolute Difference with Rectification

2. Optimized SSAD Calculation

$$SSAD(i,j,\xi) = \sum_{(s,t)\in w(i,j)} (\sum_{k=1}^{n} AD_k(s,t,\xi))$$

$$SSAD(i,j,\xi) = \sum_{s=i-m}^{i+m} (\sum_{t=j-m}^{j+m} (\sum_{k=1}^{n} AD_k(s,t,\xi)))$$

$$VSUM(i,j,\xi) = \sum_{t=j-m}^{j+m} (\sum_{k=1}^{n} AD_k(s,t,\xi))$$

$$SSAD(i,j,\xi) = SSAD(i-1,j,\xi)$$
$$-VSUM(i-m-1,j,\xi)+VSUM(i+m,j,\xi)$$

$$VSUM(i,j,\xi) = VSUM(i,j-1,\xi)$$
$$-\sum_{k=1}^{n}(AD_k(i,j-m-1,\xi))+\sum_{k=1}^{n}(AD_k(i,j+m,\xi))$$

3. Minimum Finder

Subpixel Disparity Detection –

Figure 9: Example scenes demonstrating the performance of subpixel interpolation of depth

(a) An intensity image

(b) The corresponding depth map with 30 disparity range

(c) The interpolated depth in a precision of 8 bits

A Camera Head –

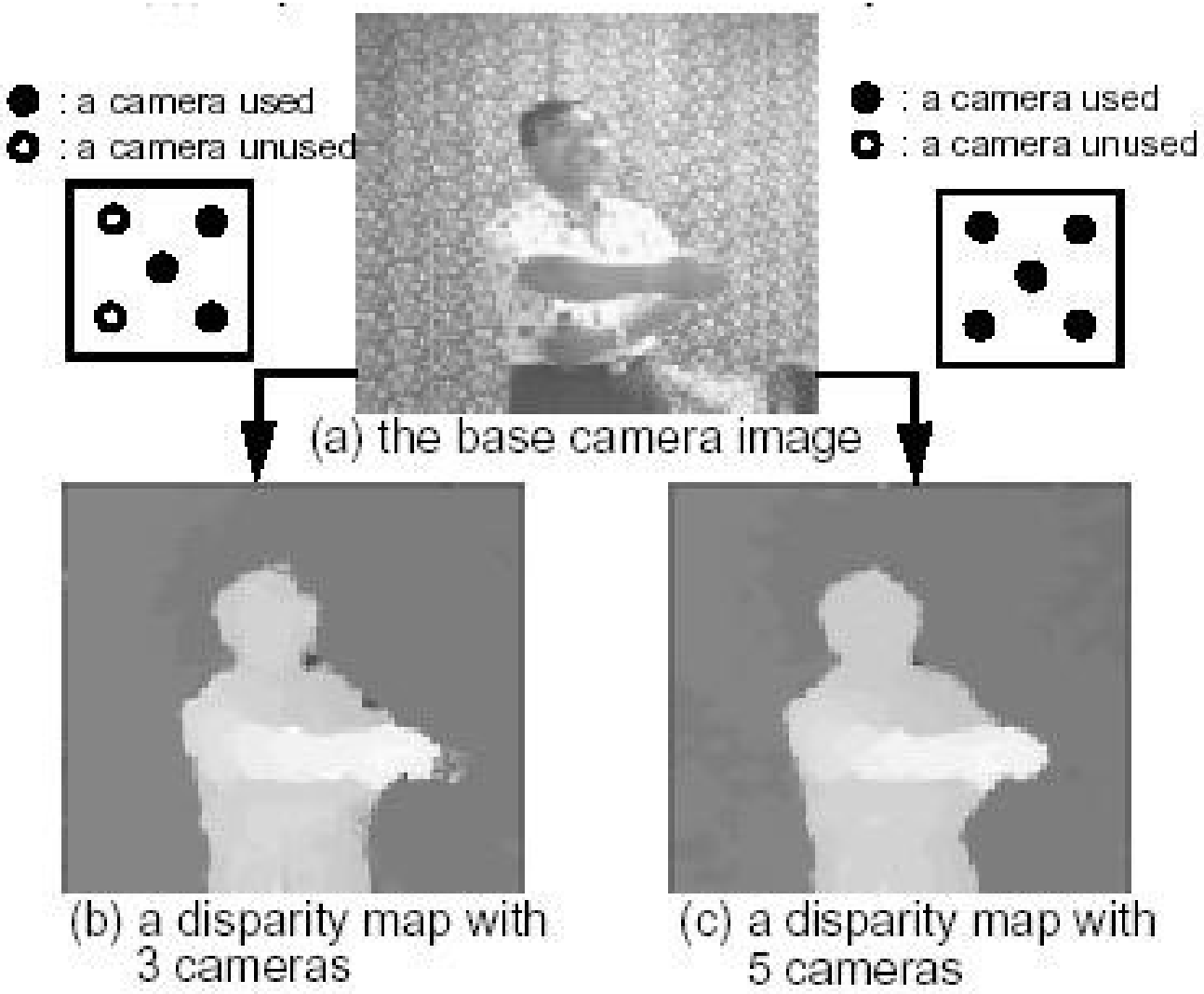


Figure 10: Example scenes of disparity map with occlusions and without occlusions

❑New Applications of the Stereo Machine:

Z Keying -

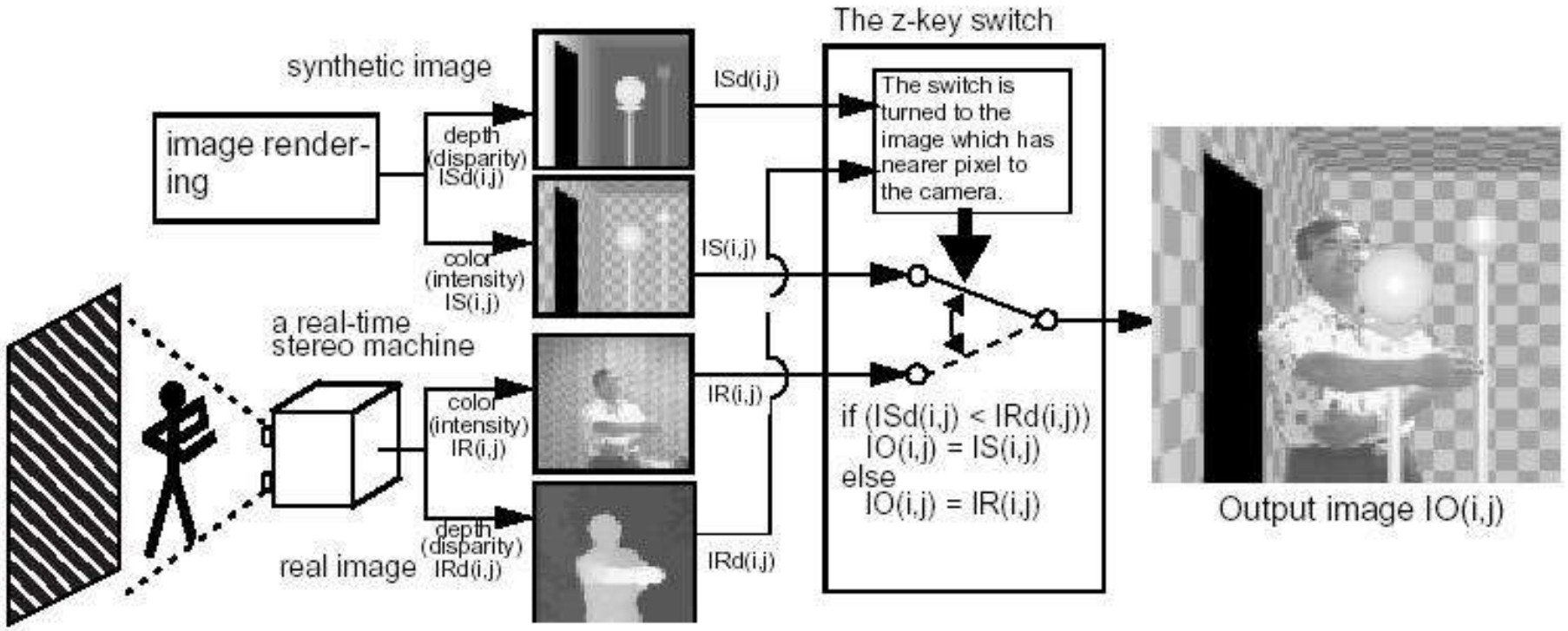
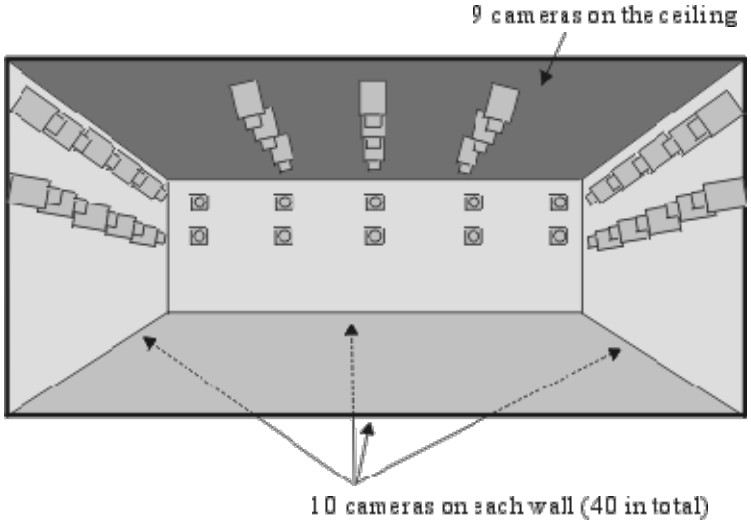


Figure 11: The scheme for z keying

Virtualized Reality -

CMU's 3D Dome



CMU's 3D Room

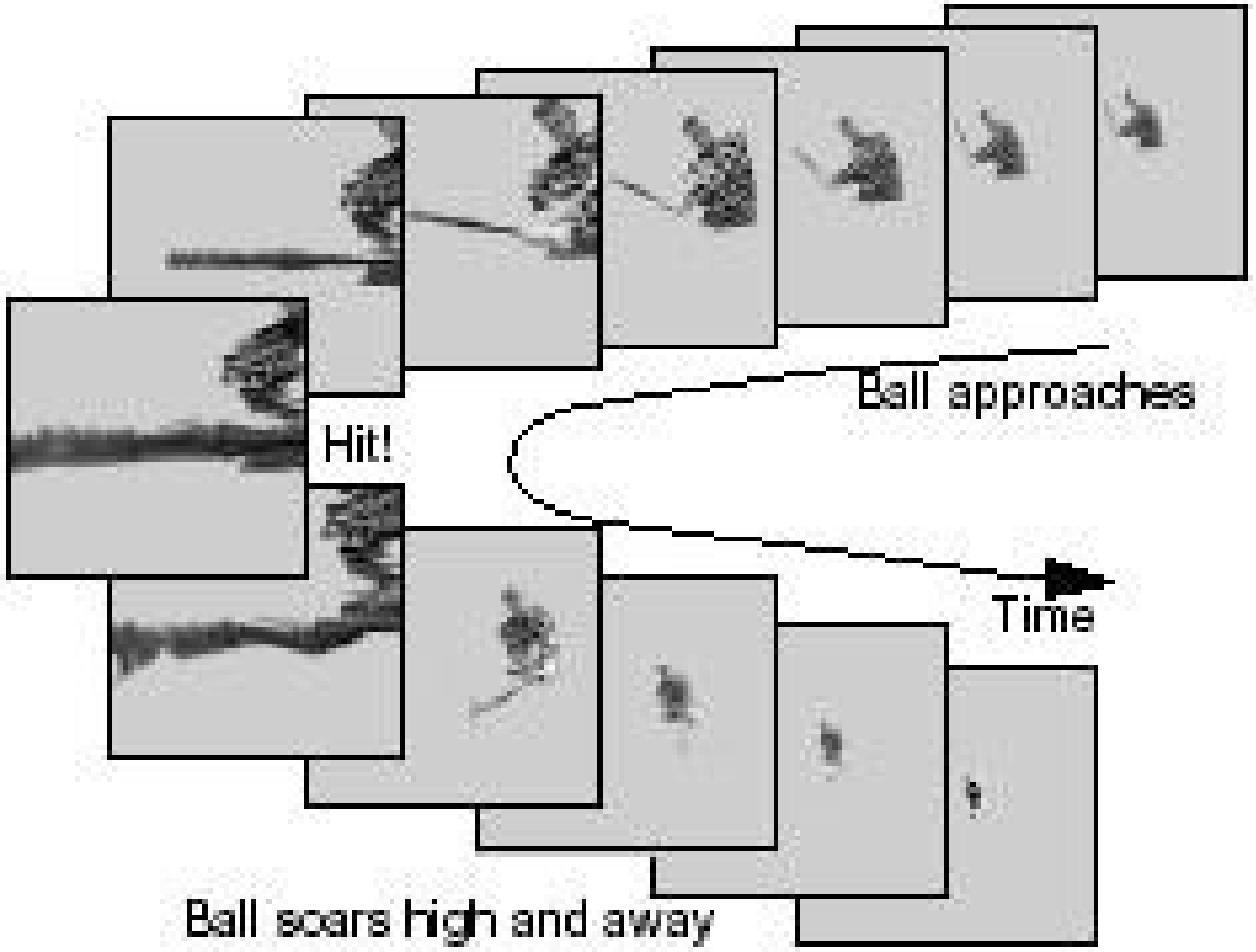
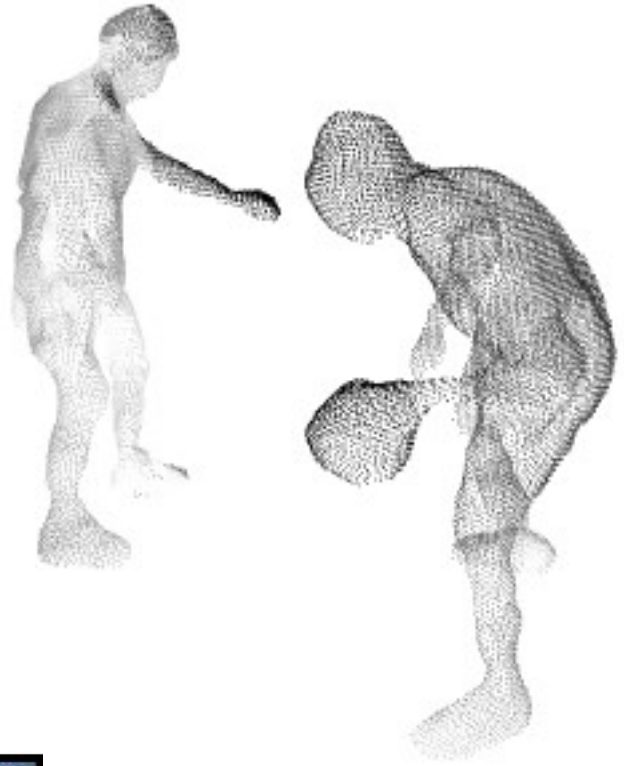


Figure 12: A "baseball" sequence from the ball's point of view



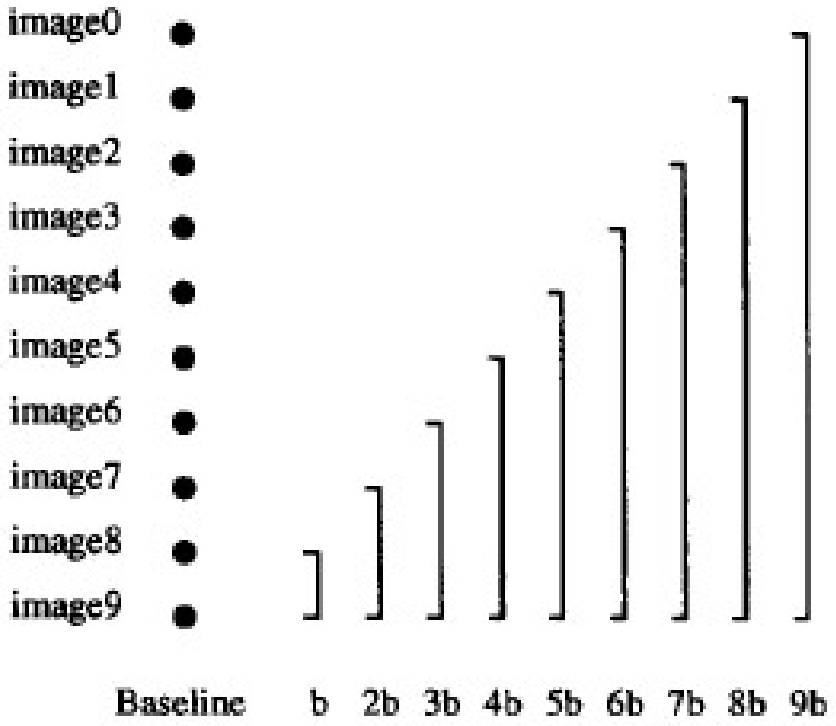
Summary of A Multiple-Baseline Stereo

- **Basic Approach**
  - Choose a reference view
  - Use your favorite stereo algorithm BUT
    - » replace two-view SSD with SSD over all baselines

- **Limitations**
  - Must choose a reference view (bad)
  - Visibility!

The Effect of Baseline on Depth Estimation
(A Multiple-Baseline Stereo: IEEE PAMI)

Figure 2: An example scene. The grid pattern in the background has ambiguity of matching.



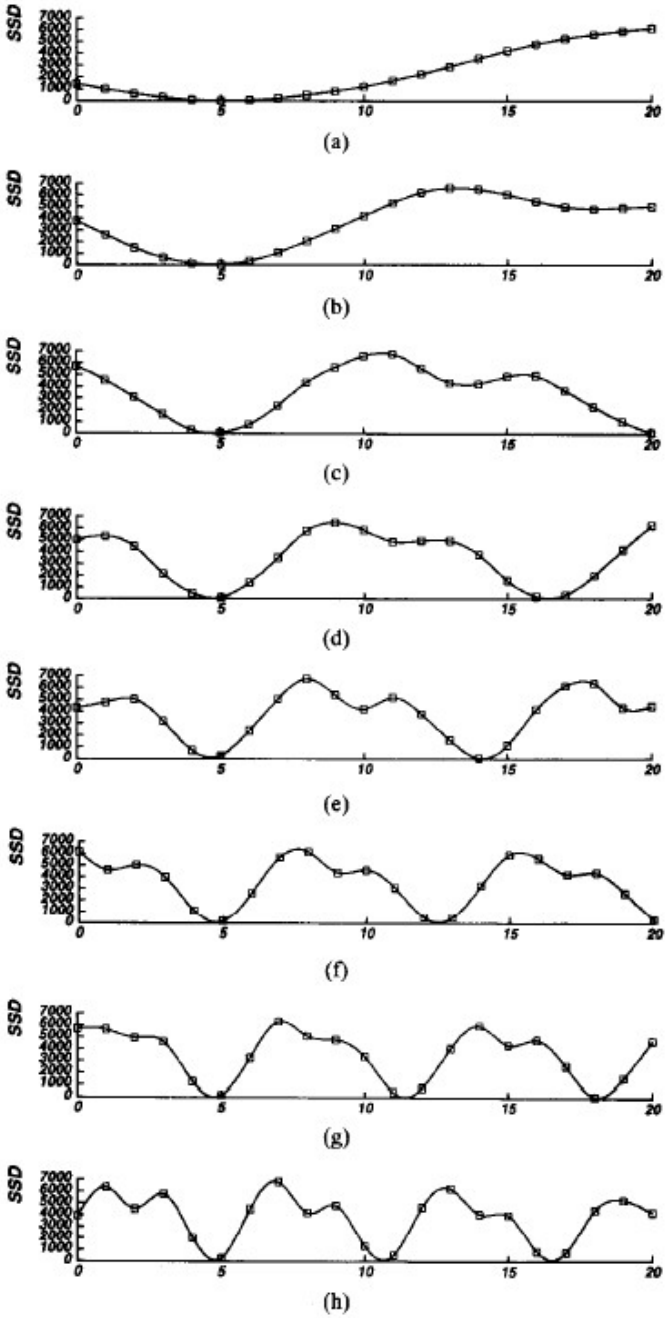


Fig. 5. SSD values versus inverse distance: (a) $B = b$; (b) $B = 2b$; (c) $B = 3b$; (d) $B = 4b$; (e) $B = 5b$; (f) $B = 6b$; (g) $B = 7b$; (h) $B = 8b$. The horizontal axis is normalized such that $8bF = 1$.

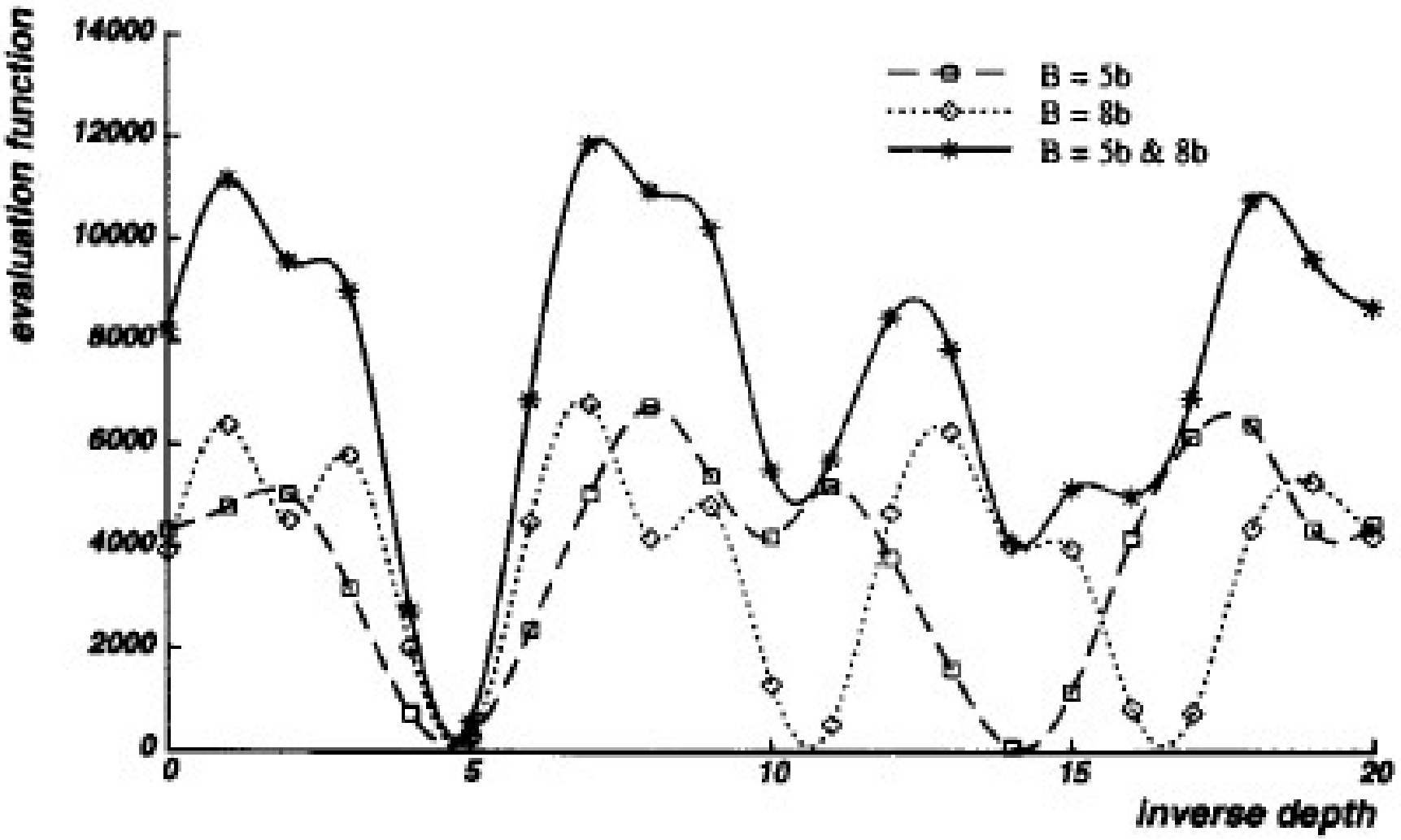


Fig. 6. Combining two stereo pairs with different baselines.

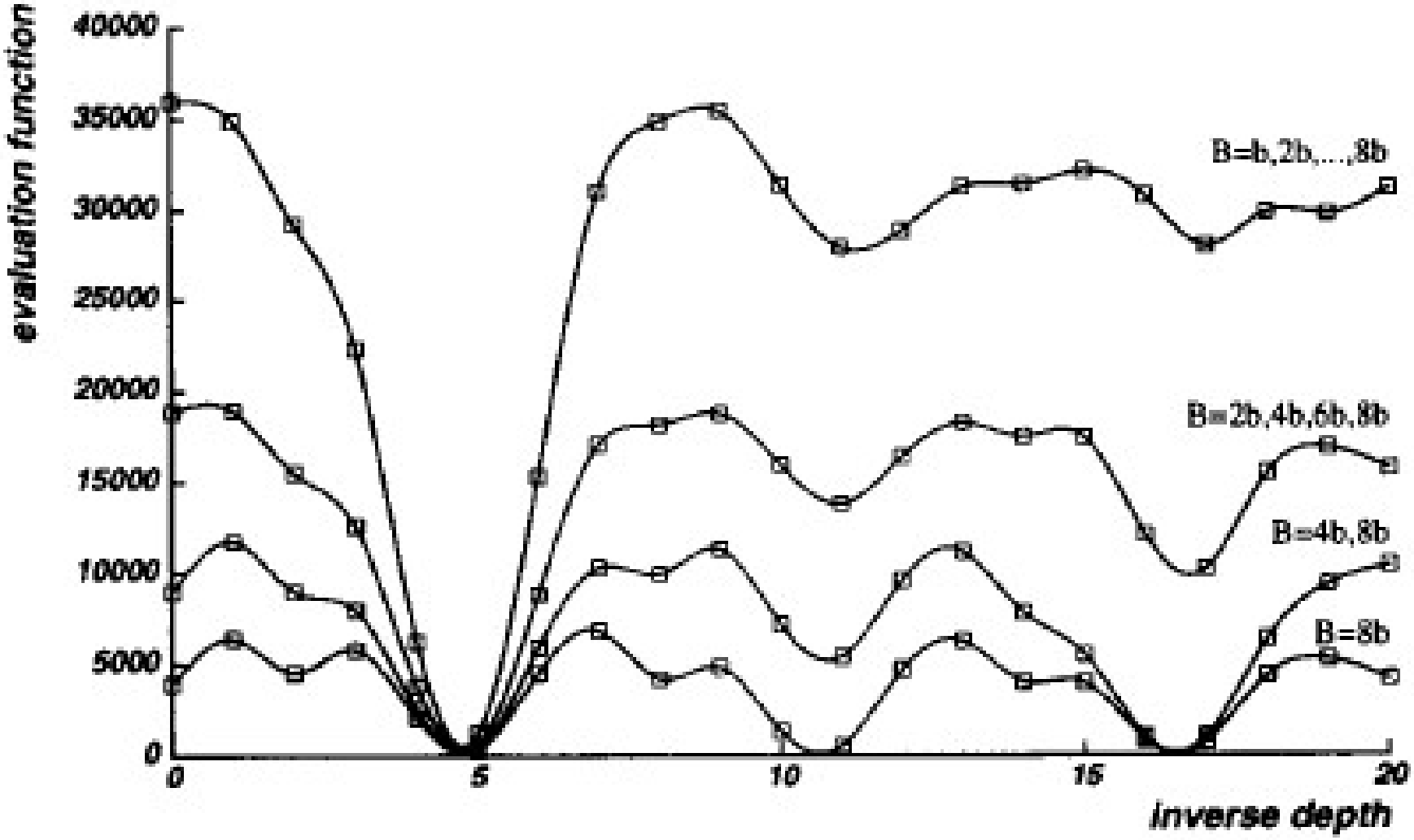
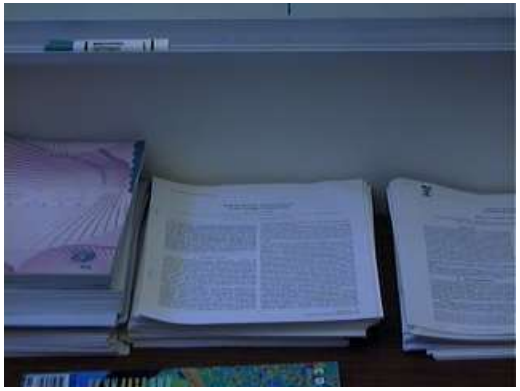


Fig. 7. Combining multiple baseline stereo pairs.

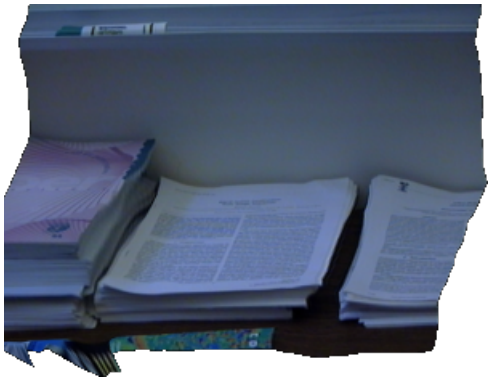
Stereo Applications

Depth From Disparity:

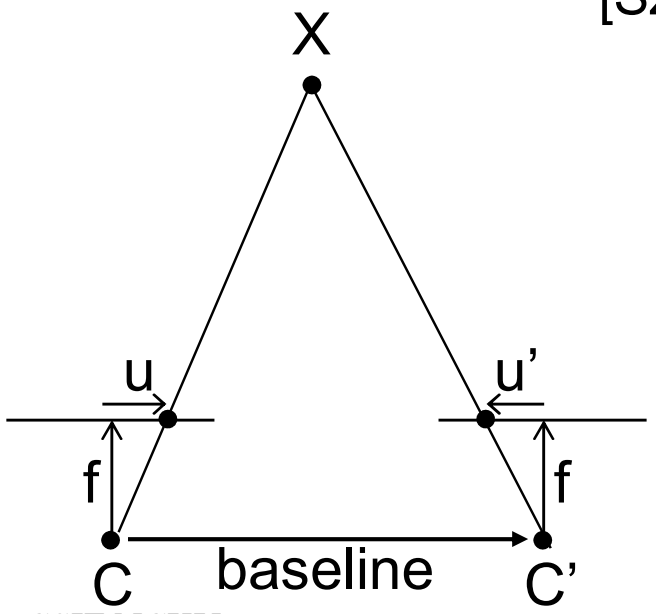


input image (1 of 2)

depth map
[Szeliski & Kang '95]



3D rendering



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

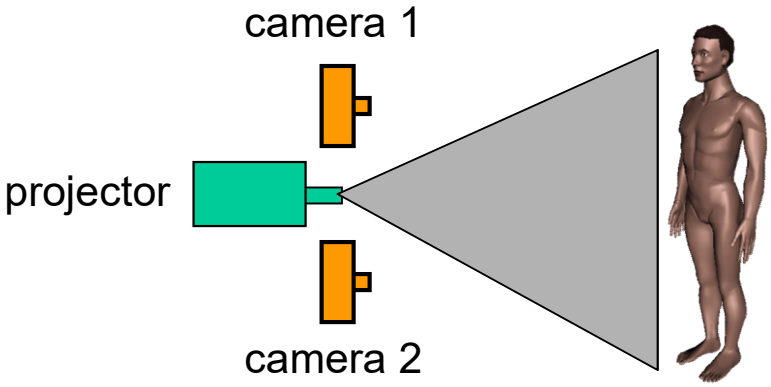
Image-Based Rendering:

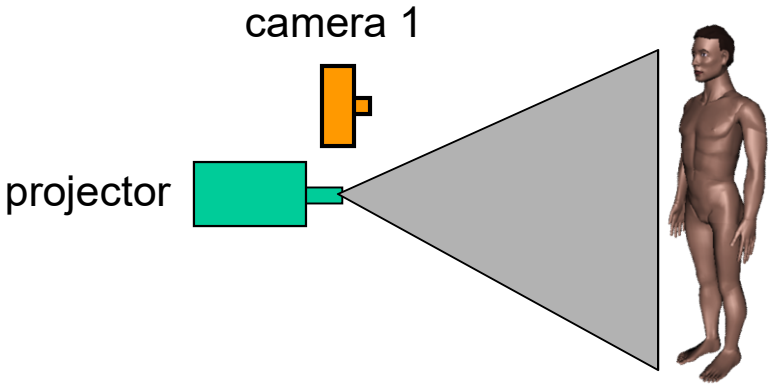
❑ Render new views from raw disparity

- S. M. Seitz and C. R. Dyer, View Morphing, *Proc. SIGGRAPH 96*, 1996, pp. 21-30.
- L. McMillan and G. Bishop. Plenoptic Modeling: An Image-Based Rendering System, *Proc. of SIGGRAPH 95*, 1995, pp. 39-46.

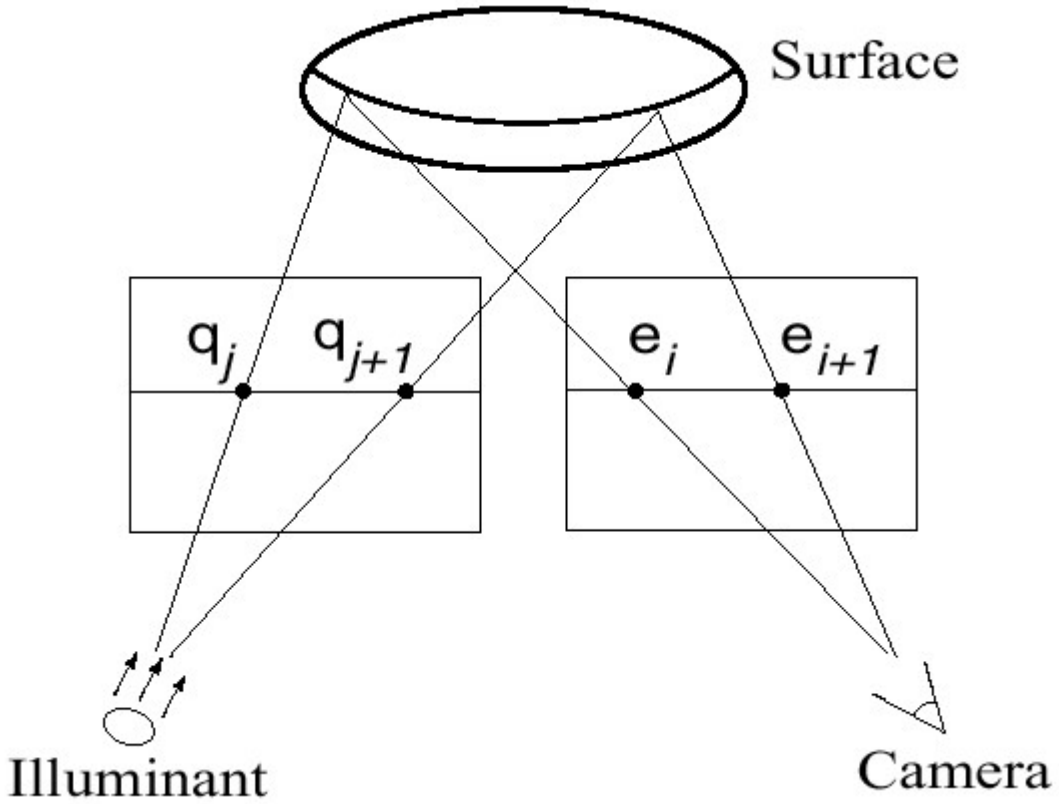
Active Stereo with Structured Light:

Li Zhang's one-shot stereo

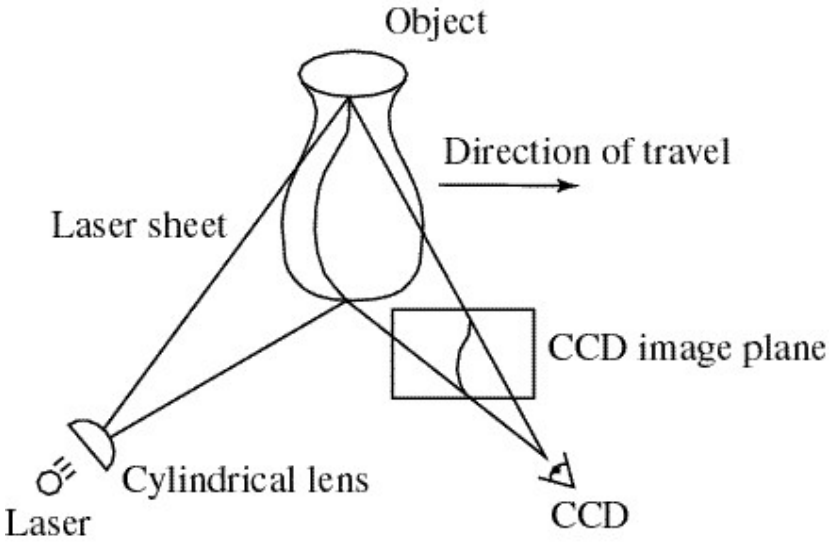


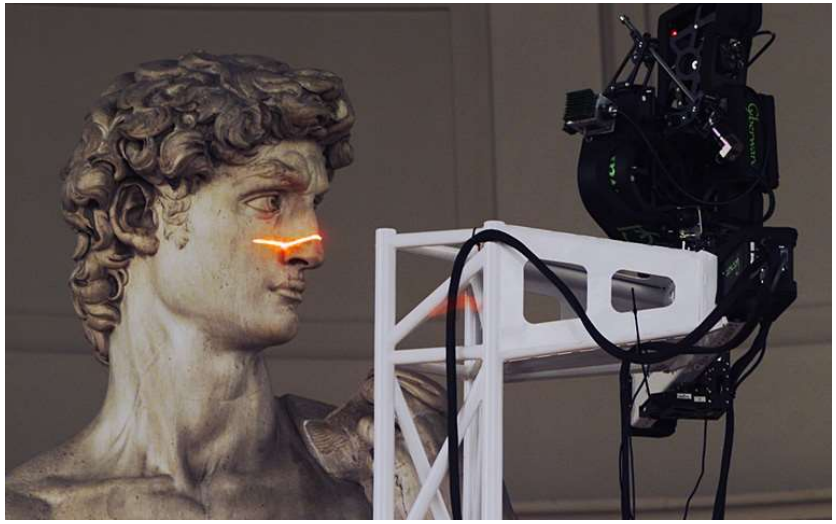


- **Project "structured" light patterns onto the object**
  - simplifies the correspondence problem


Surface
q_j
q_{j+1}
e_i
e_{i+1}
Illuminant
Camera

Structured Light Scanning - Laser Scanning:





Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/

- **Optical triangulation**
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning

Real-Time Stereo:

Nomad robot searches for meteorites in Antartica
http://www.frc.ri.cmu.edu/projects/meteorobot/index.html

real-time
stereo video

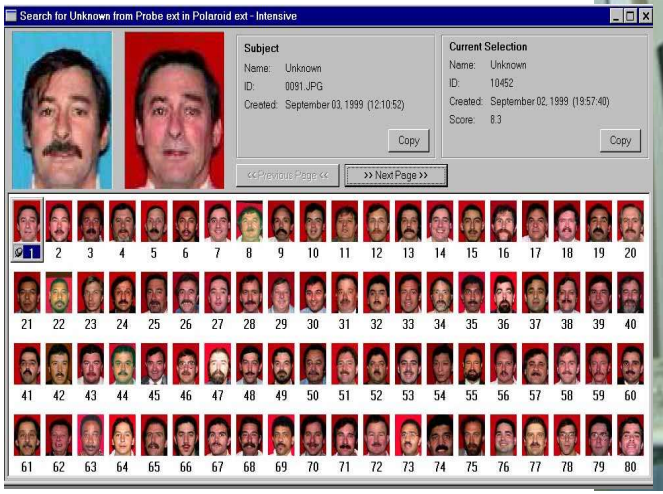
- **Used for robot navigation (and other tasks)**
  - Several software-based real-time stereo techniques have been developed (most based on simple discrete search)

Summary

- **Things to take away from this lecture**
  1) Cues for 3D inference, shape from X
  2) Epipolar geometry
  3) Stereo image rectification
  4) Stereo matching
     (1) window-based epipolar search
     (2) effect of window size
     (3) sources of error
  5) Active stereo
     (1) structured light
     (2) laser scanning

Stereo System and Database at Visionics (Project)

- **Visionics stereo system, database and demo**
  - **See previous pages – Stereo Basics**



❑Camera Calibration

SONY EVI-D30 (1/3 inch lens: 4.8 x 3.6 mm): Image resolution: w x h = 320 x 240 pixels; 1 Pixel = 0.015 mm = 0.0006 inch

Focal length at x axis: α	Focal length at y axis: β	Average focal length = (α + β)/2	The coordinates of principal point (u_0,v_0)
2652.4 Pixels	2630.8 Pixels	2641.6 Pixel = 39.6 mm = 1.58 inch	(161.7,125.7)

Coefficient of radial distortion: k1	Coefficient of radial distortion: k2	Coefficient of tangential distortion: k3	Coefficient of tangential distortion: k4
-0.273651	-0.134991	-0.006805	-0.000034

- **Z (m) = f*B/d**
  - Known: f = 2641.6 pixel and B = 15 cm, (f*B=39624)
  - Measure: d (pixel)

References

1. Olivier Faugeras, Three-Dimensional Computer Vision: A Geometric Viewpoint, Third printing, 1999, The MIT Press.
2. R.C. Gonzalez and P. Wintz, Digital Image Processing, 2nd edition, 1987.
3. R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.
4. T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka, "A Stereo Machine for Video-rate Dense Depth Mapping and Its New Applications," IEEE CVPR, June 18-20, 1996.
5. K. Konolige, "Small Vision Systems: Hardware and Implementation," Artificial Intelligence Center, SRI International, 1997.
6. M. Okutomi and T. Kanade, "A Multiple-Baseline Stereo," IEEE PAMI, Vol. 15, No. 4, April 1993.
7. Stereo benchmark website: http://vision.middlebury.edu/stereo/