CS 534: Computer Vision
Stereo Imaging

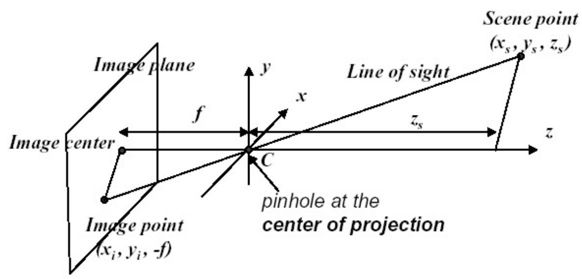
Ahmed Elgammal
Dept of Computer Science
Rutgers University

Outlines

- Depth Cues
- Simple Stereo Geometry
- Epipolar Geometry
- Stereo correspondence problem
- Algorithms

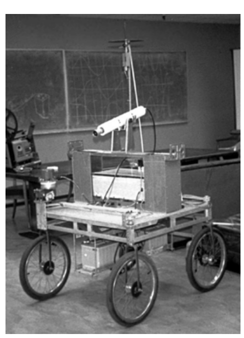
Recovering the World From Images

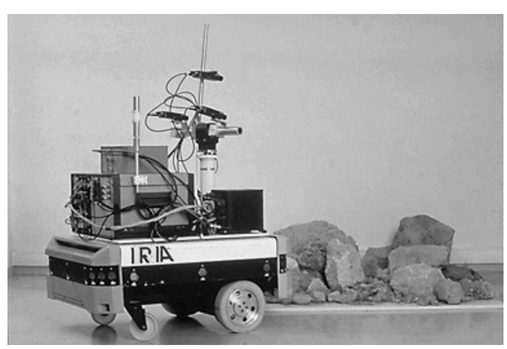
We know:
- 2D Images are projections of 3D world.
- A given image point is the projection of any world point on the *line of sight*
- So how can we recover depth information



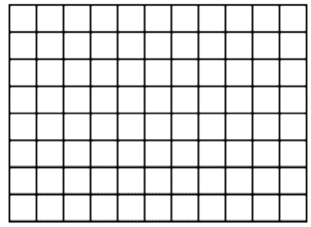
Why to recover depth ?

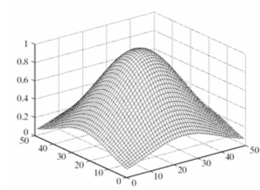
- Recover 3D structure, reconstruct 3D scene model, many computer graphics applications
- Visual Robot Navigation
- Aerial reconnaissance
- Medical applications

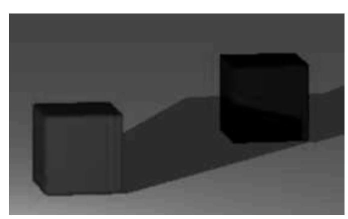

The Stanford Cart, H. Moravec, 1979.

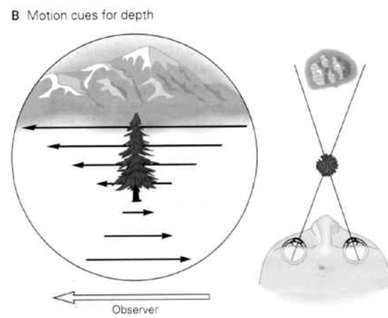


The INRIA Mobile Robot, 1990.




1
0.8
0.6
0.4
0.2
0
50
40
30
20
10
0
0
10
20
30
40
50



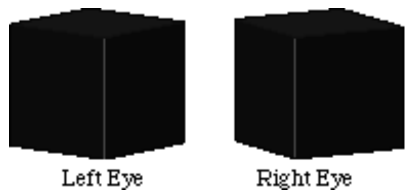


Motion parallax

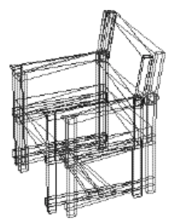
Depth Cues

- Monocular Cues
  - Occlusion – Interposition
  - Relative height: the object closer to the horizon is perceived as farther away, and the object further from the horizon is perceived as closer.
  - Familiar size: when an object is familiar to us, our brain compares the perceived size of the object to this expected size and thus acquires information about the distance of the object.
  - Texture Gradient: all surfaces have a texture, and as the surface goes into the distance, it becomes smoother and finer.
  - Shadows
  - Perspective
  - Focus
- Motion Parallax (also Monocular)
- Binocular Cues
- In computer vision: large research on shape-from-X (should be called depth from X)

Binocular Cues: stereopsis



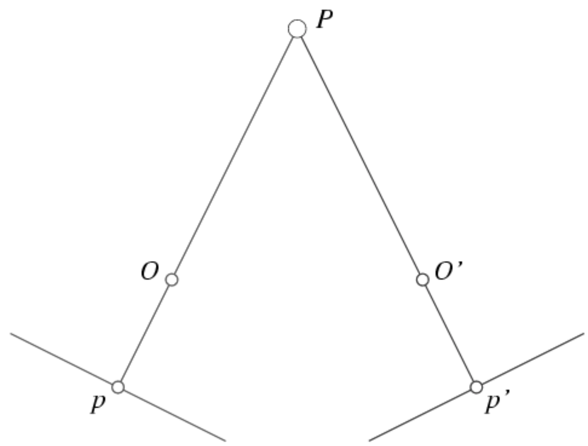
- Binocular disparity: The slight difference between the viewpoints of your two eyes is called.



Random Dot stereogram

- created by Dr. Bela Julesz (from Rutgers), described in the book Foundations of Cyclopean Perception. 1971 : The left and right images are identical except for a central square region that is displaced slightly in one of the images, when fused binocularly, the images yields the impression of a central square floating in front of the background.

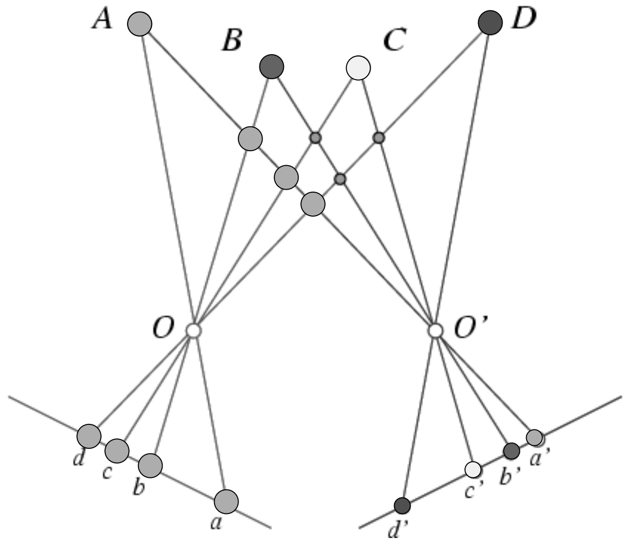
- Given multiple views we can recover scene point - Triangulation



Stereo vision involves two processes:

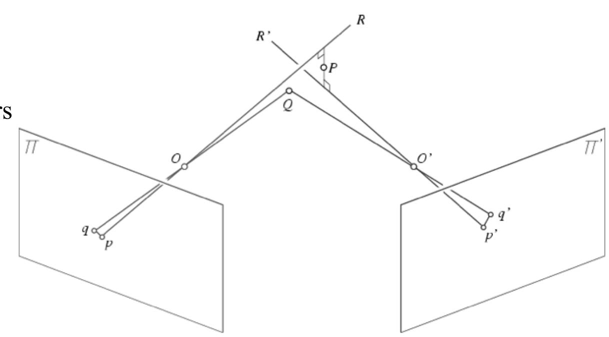
- *Fusion* of features observed by two or more cameras: which point corresponds to which point ?
- *Reconstruction* of 3D preimage: how to intersect the rays.

(Binocular) Fusion



Reconstruction

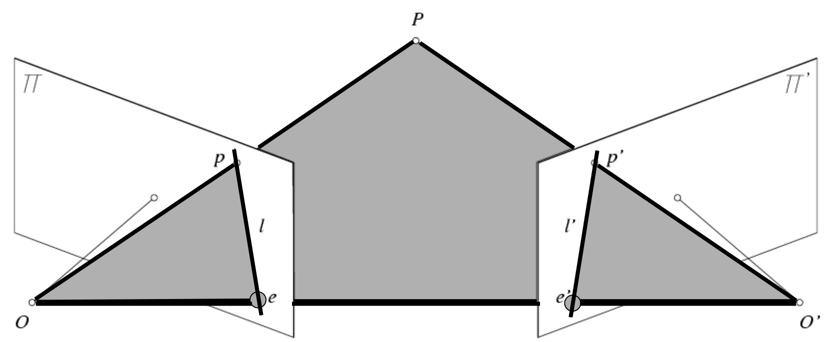
In practice rays never intersect:
• calibration errors
• feature localization errors



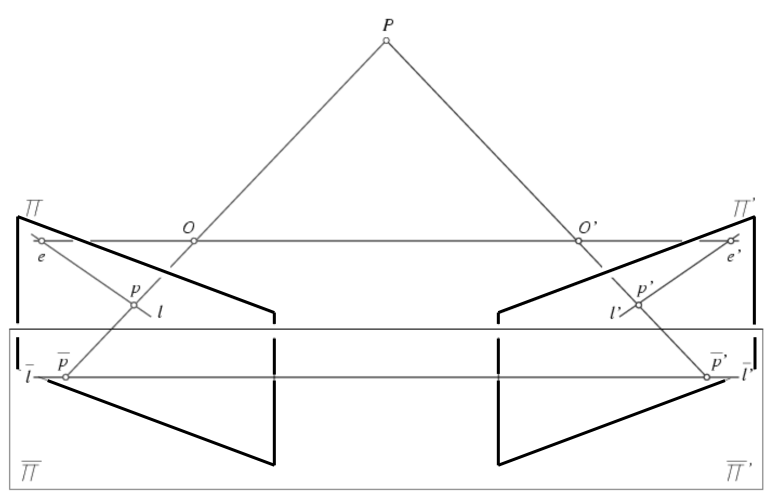
•Algebraic linear method: four equations in three unknown – use linear least squares to find *P*

• Non-Linear Method: find *Q* minimizing $d^2(p,q) + d^2(p',q')$

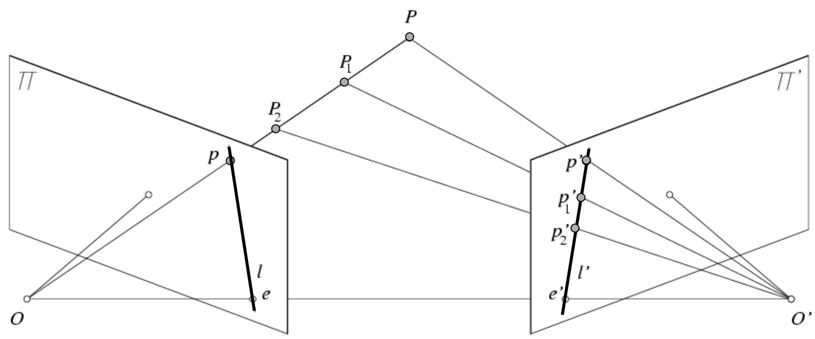
Epipolar Geometry



- Epipolar Plane
- Baseline
- Epipoles
- Epipolar Lines



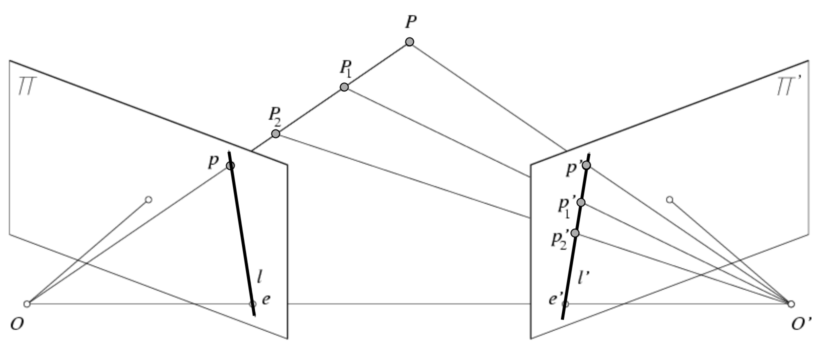
Epipolar Constraint



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

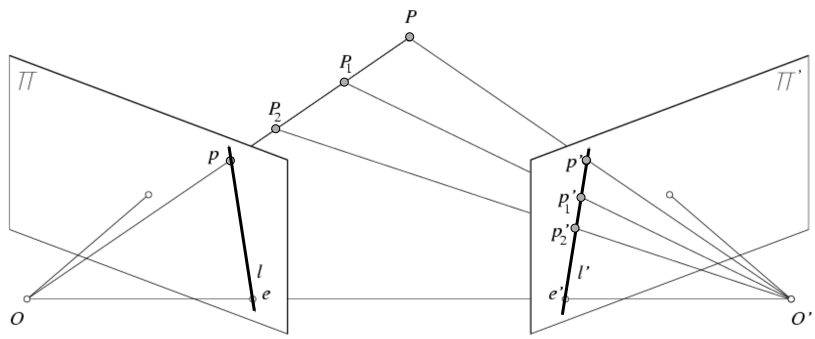
- Potential matches for *p'* have to lie on the corresponding epipolar line *l*.

Epipolar Constraint



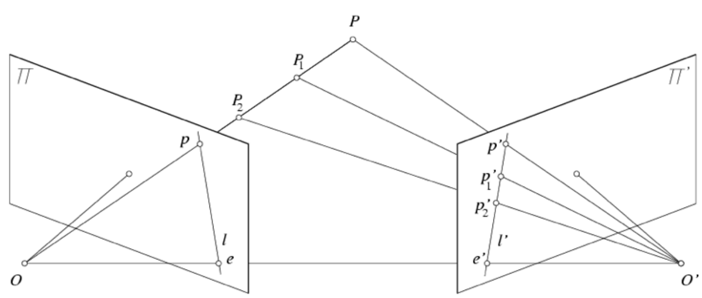
- First scene point possibly corresponding to *p* is *O* : (any point closer to the left image than O would be between the lens and the image plane, and could not be seen.)
- So, first possible corresponding point in the right image is *e'*

Epipolar Constraint



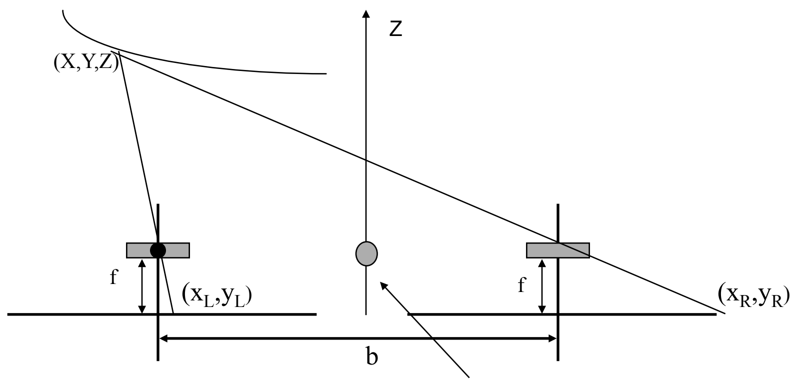
• Last scene point possibly corresponding to *p* is point at infinity along *p* line of sight
• but its image is the vanishing point of the ray *Op* in the right camera
• so we know two points on the epipolar line, any corresponding point *p'* is between *e'* and this vanishing point

Epipolar Constraint



epipole *e'*
• this is image of the left lens center in the right image
• this point O lies on the line of sight for every point in the left image
• All epipolar lines for all points in the left image must pass through *e'*
• might not be in the finite field of view
Special case: image planes parallel to the baseline (standard stereo sitting):
• epipolar lines are scan lines
• epipoles at infinity

Standard Stereo imaging

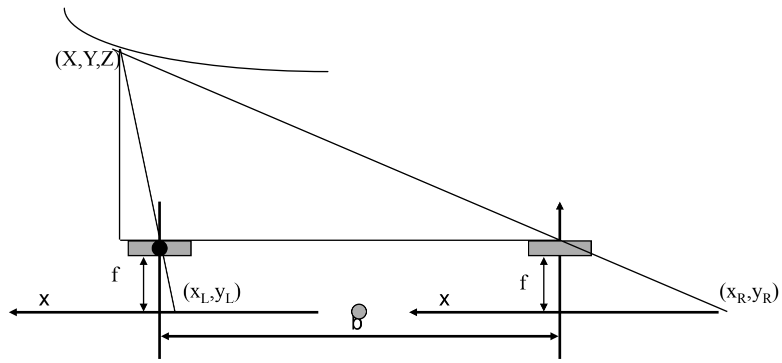


- Optical axes are parallel
- Optical axes separated by **baseline**, b.
- Line connecting lens centers is perpendicular to the optical axis, and the x axis is parallel to that line
- 3D coordinate system is a **cyclopean** system centered between the cameras

Stereo imaging

- (X,Y,Z) are the coordinates of P in the Cyclopean coordinate system.
- The coordinates of P in the left camera coordinate system are $(X_L,Y_L,Z_L) = (X+b/2, Y, Z)$
- The coordinates of P in the right camera coordinate system are $(X_R,Y_R,Z_R) = (X-b/2, Y, Z)$
- So, the x image coordinates of the projection of P are
  - $x_L = (X+b/2)f/Z$
  - $x_R = (X-b/2)f/Z$
- Subtracting the second equation from the first, and solving for Z we obtain:
  - $Z = bf/(x_L - x_R)$
- We can also solve for X and Y:
  - $X = b(x_L + x_R)/2(x_L-x_R)$
  - $Y = by/(x_L-x_R)$

Stereo imaging

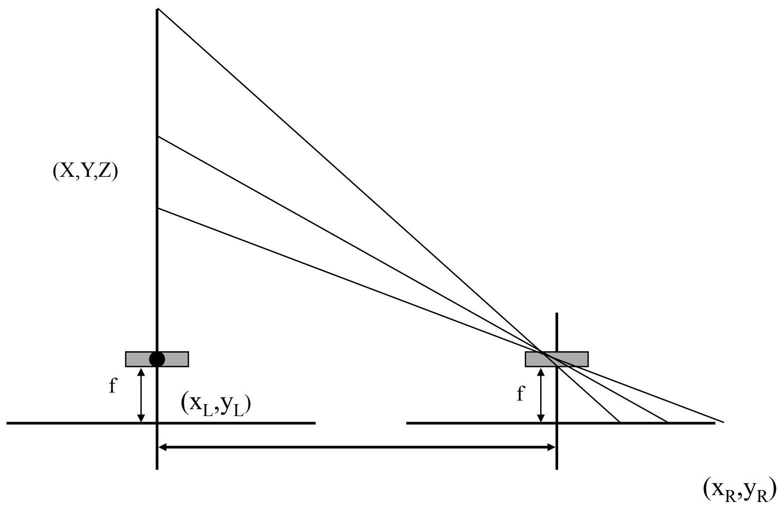


- x_L - x_R is called the **disparity**, d, and is always negative
- X= $(b[x_R + x_L]/2)/ d$ Y = by/ d Z = bf/d

Stereo imaging

- Z = bf/d
- Depth is inversely proportional to |disparity|
  - disparity of 0 corresponds to points that are infinitely far away from the cameras
  - in digital systems, disparity can take on only integer values (ignoring the possibility of identifying point locations to better than a pixel resolution)
  - so, a disparity measurement in the image just constrains distance to lie in a given range
- Disparity is directly proportional to b
  - the larger b, the further we can accurately range
  - but as b increases, the images decrease in common field of view

Range versus disparity



Stereo imaging

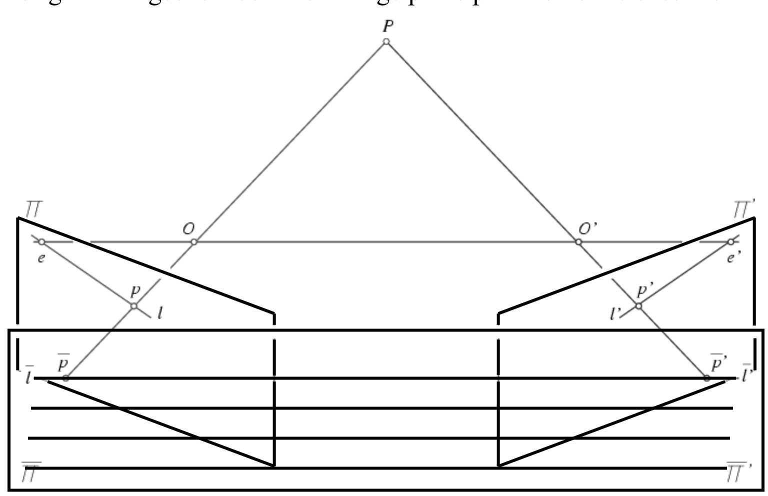
- Definition: A scene point, P, visible in both cameras gives rise to a pair of image points called a **conjugate pair**.
  - the conjugate of a point in the left (right) image must lie on the same image row (line) in the right (left) image because the two have the same y coordinate
  - this line is called the **conjugate line = epipolar line**.
  - so, for our simple image geometry, all epipolar lines are parallel to the x axis

A more practical stereo image model

- Difficult, practically, to
  - have the optical axes parallel
  - have the baseline perpendicular to the optical axes
- Also, we might want to tilt the cameras towards one another to have more overlap in the images
- Calibration problem - finding the transformation between the two cameras
  - it is a rigid body motion and can be decomposed into a rotation, **R**, and a translation, **T**.

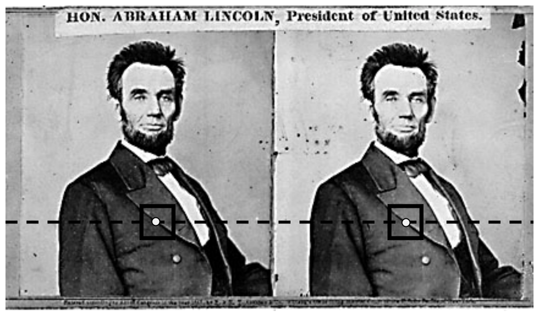
Image Rectification

Project original images to a common image plane parrallel to the baseline



All epipolar lines are parallel in the rectified image plane.

Your basic stereo algorithm



For each epipolar line

For each pixel in the left image

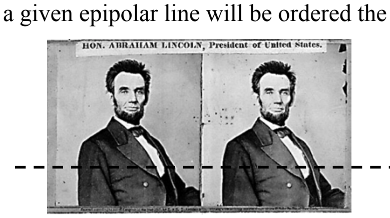
- compare with every pixel on same epipolar line in right image
- pick pixel with minimum match cost

Improvement: match ***windows***

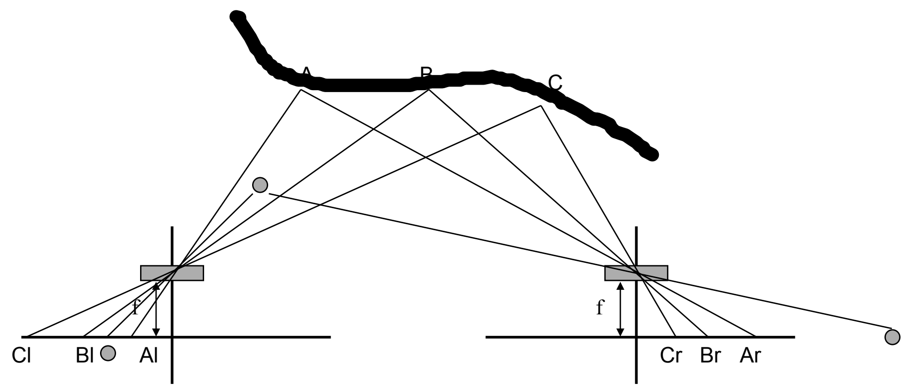
- This should look familiar...

Stereo correspondence problem

- Given a point, p, in the left image, find its conjugate point in the right image
  - called the stereo correspondence problem
  - Different approaches
- What constraints simplify this problem?
  - Epipolar constraint - need only search for the conjugate point on the epipolar line
  - Negative disparity constraint - need only search the epipolar line to the “right” of the vanishing point in the right image of the ray through p in the left coordinate system
  - Continuity constraint - if we are looking at a continuous surface, images of points along a given epipolar line will be ordered the same way



Continuity constraint

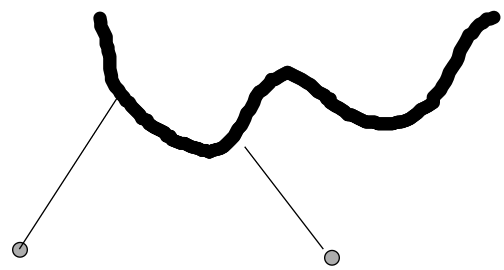


Stereo correspondence problem

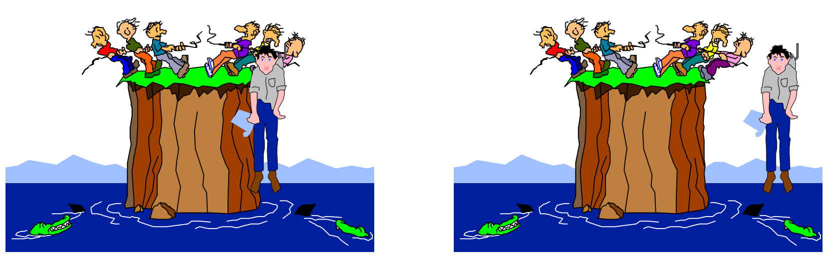
- Similarity of correspondence functions along adjacent epipolar lines
- Disparity gradient constraint - disparity changes slowly over most of the image.
  - Exceptions occur at and near occluding boundaries where we have either discontinuities in disparity or large disparity gradients as the surface recedes away from sight.

Why is the correspondence problem hard

- Occlusion
  - Even for a smooth surface, there might be points visible in one image and not the other
  - Consider aerial photo pair of urban area - vertical walls of buildings might be visible in one image and not the other
  - scene with depth discontinuities (lurking objects) violates continuity constraint and introduces occlusion



Why is the correspondence problem hard?



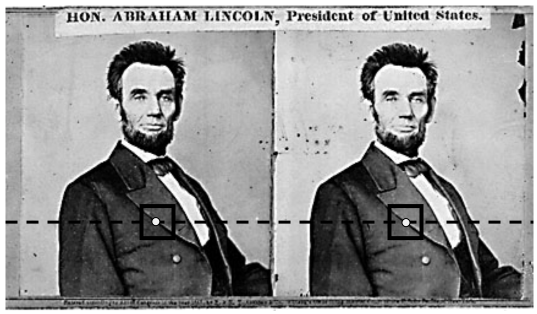
Why is the correspondence problem hard?

- Variations in intensity between images due to
  - noise
  - specularities
  - shape-from-shading differences
- Coincidence of edge and epipolar line orientation
  - consider problem of matching horizontal edges in an ideal left right stereo pair
  - will obtain good match all along the edge
  - so, edge based stereo algorithms only match edges that cross the epipolar lines

Approaches to Find Correspondences

- Intensity Correlation-based approaches
- Edge / feature matching approaches
- Dynamic programming
- Energy minimization / Graph cuts
- Probabilistic approaches

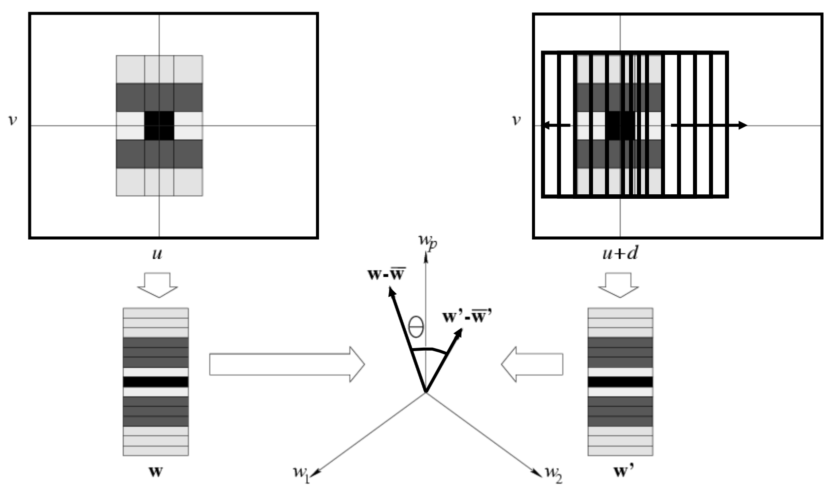
Your basic stereo algorithm



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 familiar...

Correlation Methods (1970--)

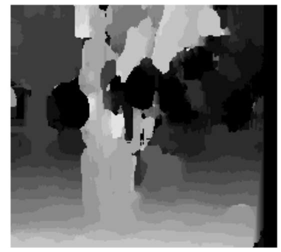


Slide the window along the epipolar line until *w.w'* is maximized.

Normalized Correlation: minimize θ instead. ⟺ Minimize $|w-w'|^2$.

Window size

W = 3



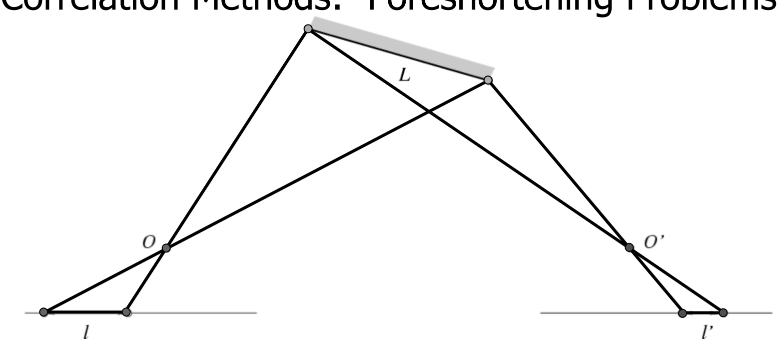
W = 20

- Effect of window size
  - Smaller window
    - \+ More details
    - \- More noise
  - Larger window
    - \+ Less noise
    - \- Less details

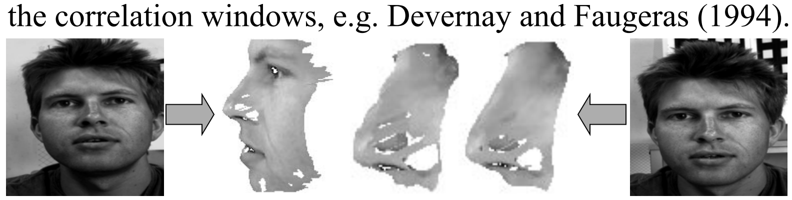
Better results with *adaptive window*

- 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

Correlation Methods: Foreshortening Problems

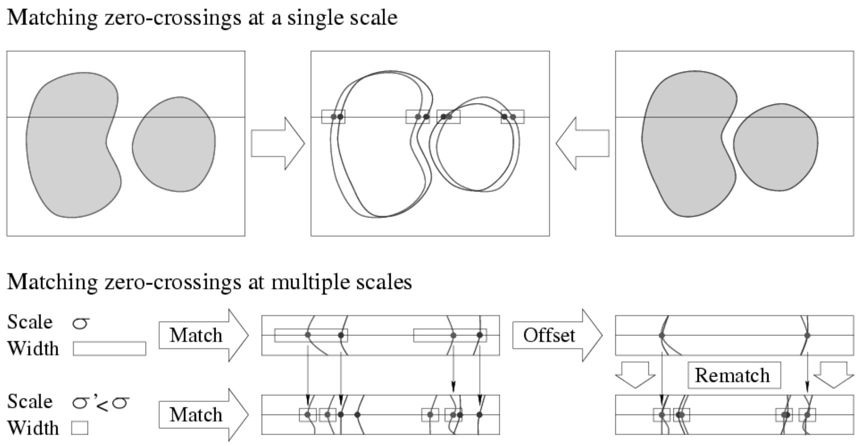


Solution: add a second pass using disparity estimates to warp the correlation windows, e.g. Devernay and Faugeras (1994).



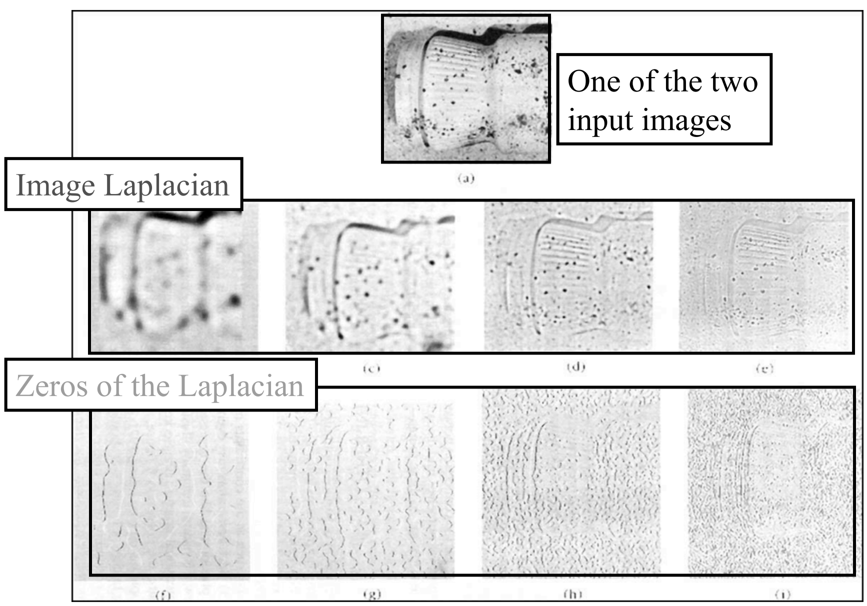
Reprinted from "Computing Differential Properties of 3D Shapes from Stereopsis without 3D Models," by F. Devernay and O. Faugeras, Proc. IEEE Conf. on Computer Vision and Pattern Recognition (1994). © 1994 IEEE.

Multi-Scale Edge Matching (Marr, Poggio and Grimson, 1979-81)

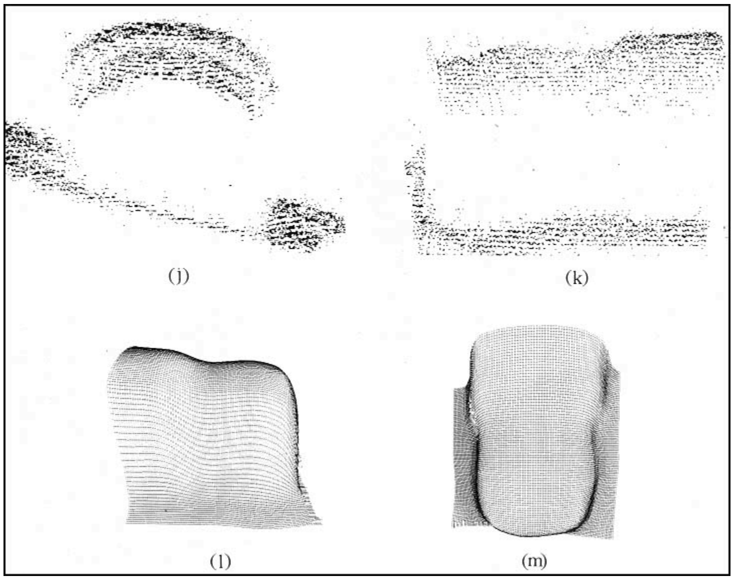


- Edges are found by repeatedly smoothing the image and detecting the zero crossings of the second derivative (Laplacian).
- Matches at coarse scales are used to offset the search for matches at fine scales (equivalent to eye movements).

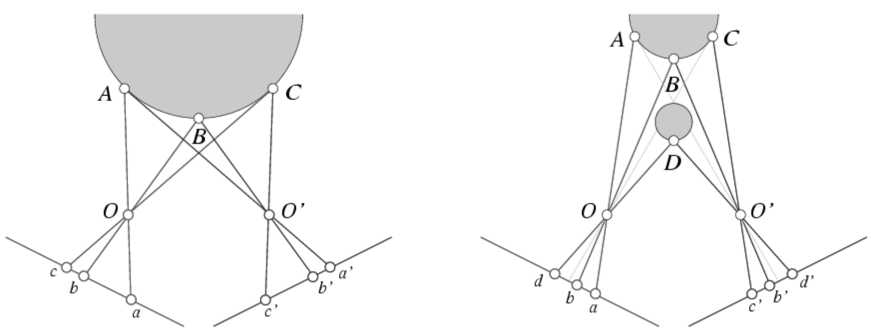
Multi-Scale Edge Matching (Marr, Poggio and Grimson, 1979-81)



Multi-Scale Edge Matching (Marr, Poggio and Grimson, 1979-81)



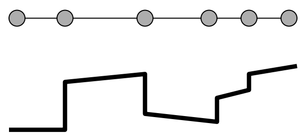
The Ordering Constraint



In general the points are in the same order on both epipolar lines.

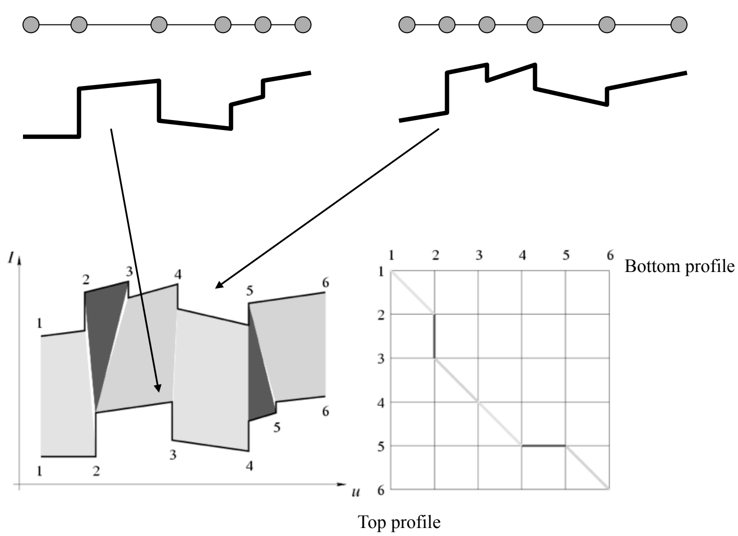
But it is not always the case..

Dynamic Programming (Baker and Binford, 1981)

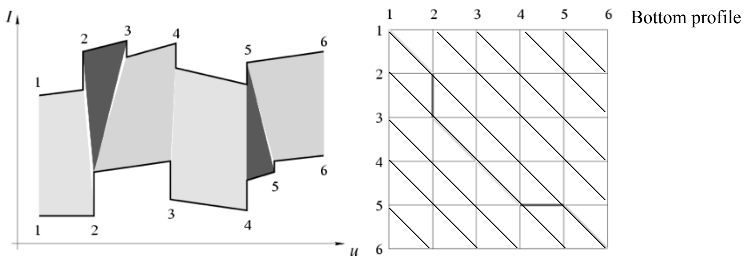


- Assume a set of feature points have been found.
- Match the intervals separating those points along the intensity profiles
- Keep the order : the order of the feature points must be the same

Dynamic Programming (Baker and Binford, 1981)



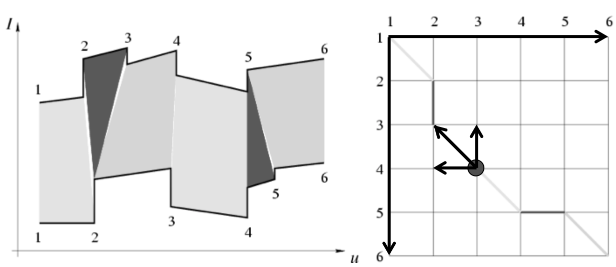
Dynamic Programming (Baker and Binford, 1981)



Find the minimum-cost path going monotonically down and right from the top-left corner of the graph to its bottom-right corner.

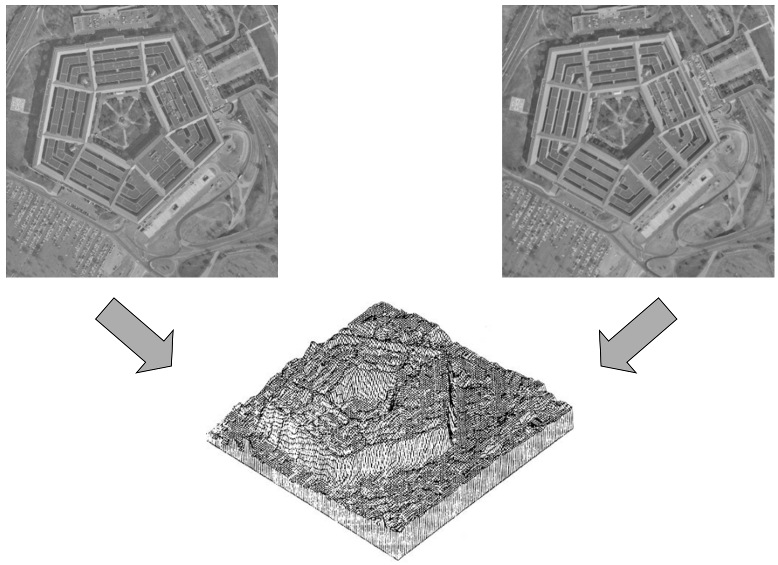
- Nodes = matched feature points (e.g., edge points).
- Arcs = matched intervals along the epipolar lines.
- Arc cost = discrepancy between intervals.

Dynamic Programming (Baker and Binford, 1981)



```
% Loop over all nodes (k, l) in ascending order.
for k = 1 to m do
  for l = 1 to n do
    % Initialize optimal cost C(k, l) and backward pointer B(k, l).
    C(k, l) ← +∞; B(k, l) ← nil;
    % Loop over all inferior neighbors (i, j) of (k, l).
    for (i, j) ∈ Inferior − Neighbors(k, l) do
      % Compute new path cost and update backward pointer if necessary.
      d ← C(i, j) + Arc − Cost(i, j, k, l);
      if d < C(k, l) then C(k, l) ← d; B(k, l) ← (i, j) endif;
    endfor;
  endfor;
endfor;
% Construct optimal path by following backward pointers from (m, n).
P ← {(m, n)}; (i, j) ← (m, n);
while B(i, j) ≠ nil do (i, j) ← B(i, j); P ← {(i, j)} ∪ P endwhile.
```

Dynamic Programming (Ohta and Kanade, 1985)



Reprinted from "Stereo by Intra- and Intet-Scanline Search," by Y. Ohta and T. Kanade, IEEE Trans. on Pattern Analysis and Machine Intelligence, 7(2):139-154 (1985). © 1985 IEEE.

Approaches to Find Correspondences

- Intensity Correlation-based approaches
  - (+) dense disparity (disparity at each pixel)
  - (-) foreshortening
    - Solution: warp windows ?
- Edge / feature matching approaches
  - (+) solve the foreshortening problem
  - (-) sparse disparity
    - Solution: interpolate intermediate disparities.
  - (-) requires feature detection
- Dynamic programming
  - (+) use both features and intensities
  - (-) emphasize ordering constraint
- Energy minimization / Graph cuts
- Probabilistic approaches

Results with window search

Window-based matching
(best window size)

Ground truth

From Slides by S. Seitz - University of Washington

Better methods exist...

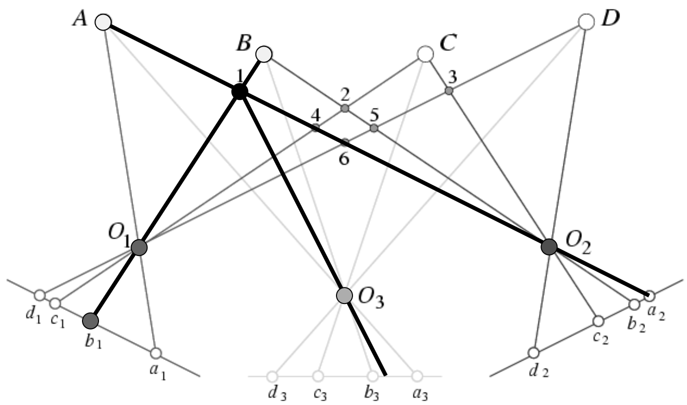
State of the art method

Ground truth

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

From Slides by S. Seitz - University of Washington

Three Views



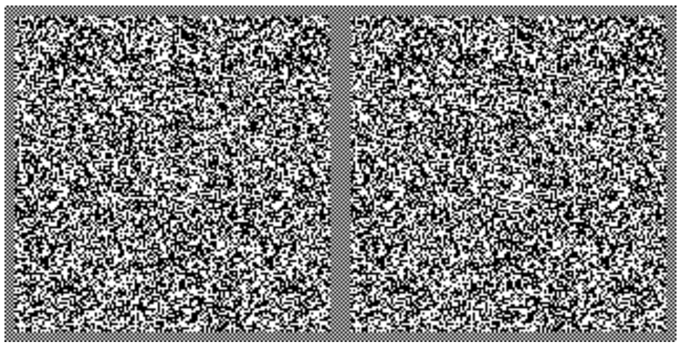
The third eye can be used for verification..

Stereo reconstruction pipeline

- Steps
  - Calibrate cameras
  - Rectify images
  - Compute disparity
  - Estimate depth
- What will cause errors?
  - Camera calibration errors
  - Poor image resolution
  - Occlusions
  - Violations of brightness constancy (specular reflections)
  - Large motions
  - Low-contrast image regions

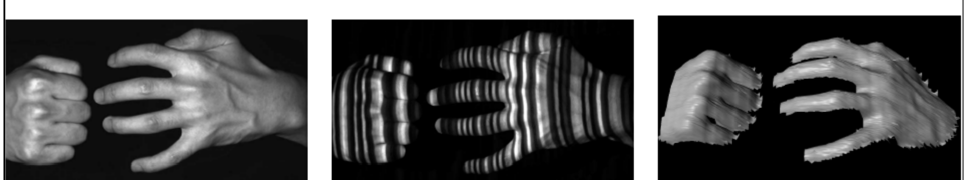
Stereo matching

- Features vs. Pixels?
  - Do we extract features prior to matching?

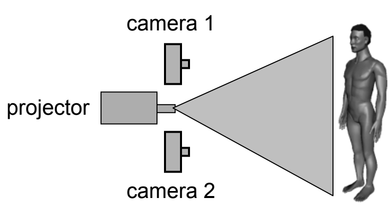


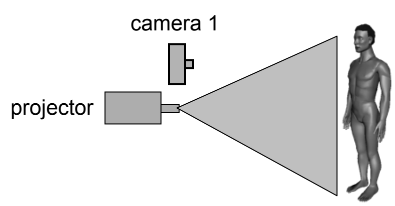
Julesz-style Random Dot Stereogram
The left and right images are identical except for a central square region that is displaced slightly in one of the images, when fused binocularly, the images yields the impression of a central square floating in front of the background.

Active stereo with structured light



Li Zhang's one-shot stereo

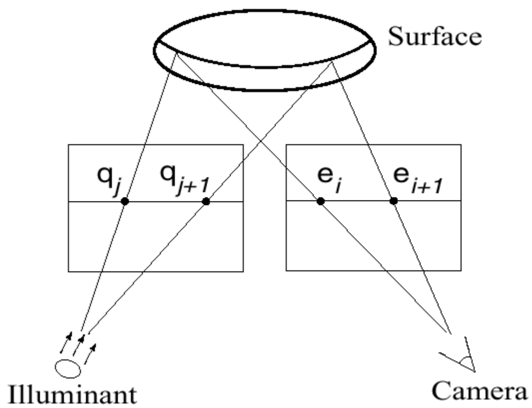




- Project "structured" light patterns onto the object
  - simplifies the correspondence problem

From Slides by S. Seitz - University of Washington

Active stereo with structured light

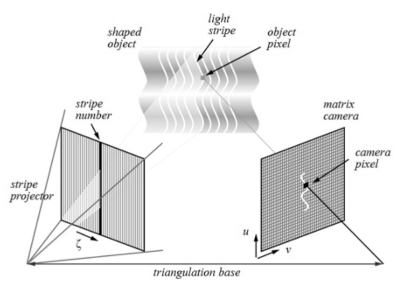


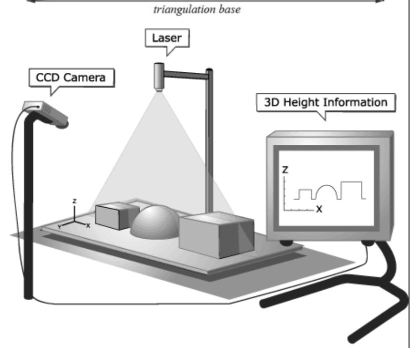
From Slides by S. Seitz - University of Washington

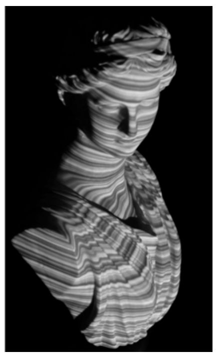
CS 534 – Stereo Imaging - 76

Structured Light 3D Scanner

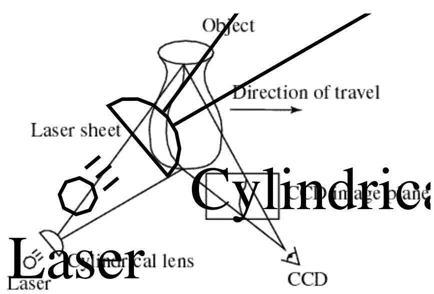
- Projector and Camera System
- Projector projects a known pattern (line or plane) of pixels.
- The camera looks at the shape of the line and uses a technique similar to triangulation to calculate the distance of every point on the line.
- The projector is typically a LCD or LCOS
- Demo

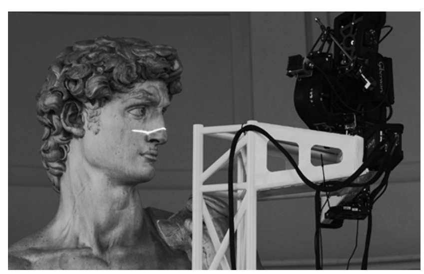






Laser scanning





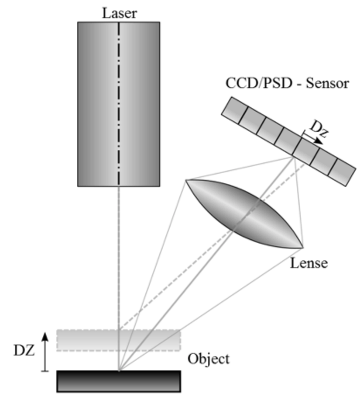
Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/

- 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

From Slides by S. Seitz - University of Washington

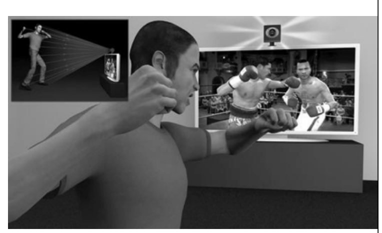
Non-contact - Active - Triangulation

- Shines a laser on the object and exploits a camera to look for the location of the laser dot
- Depending on how far away the laser strikes a surface, the laser dot appears at different places in the camera's field of view.
- the laser dot, the camera and the laser emitter form a triangle and hence it is called triangulation
  - The length of one side of the triangle, the distance between the camera and the laser emitter is known.
  - The angle of the laser emitter corner is also known.
  - The angle of the camera corner can be determined by looking at the location of the laser dot in the camera's field of view.



Time of Flight Cameras – Zcam

- Time of flight cameras capture the whole scene at the same time
  - **Illumination unit:** It illuminates the scene. Only LEDs or laser diodes are feasible as the light has to be modulated with high speeds up to 100 MHz. infrared light is used to make the illumination unobtrusive.
  - **Optics:** A lens gathers the reflected light and images the environment onto the image sensor. An optical band pass filter only passes the light with the same wavelength as the illumination unit. This helps suppress background light.
  - **Image sensor:** This is the heart of the TOF camera. Each pixel measures the time the light has taken to travel from the illumination unit to the object and back.



Time of Flight Cameras

- **Driver electronics:** Both the illumination unit and the image sensor have to be controlled by high speed signals. These signals have to be very accurate to obtain a high resolution.

- **Computation/Interface:** The distance is calculated directly in the camera. To obtain good performance, some calibration data is also used. The camera then provides a distance image over a USB or Ethernet interface.
- Demo


FOTONIC-B70 by Fotonic


SwissRanger 4000 by MESA Imaging


PMD[vision] CamCube by PMDTechnologies


USB-powered TOF camera out of the European ARTTS project


USB-powered single board PMD camera

Sources

- Forsyth and Ponce, Computer Vision a Modern approach: chapters 10,11.
- Slides by J. Ponce @ UIUC
- Slides by L.S. Davis @ UMD
- Slides by S. Seitz - University of Washington