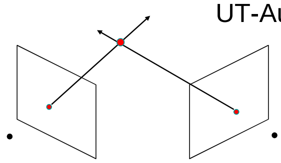
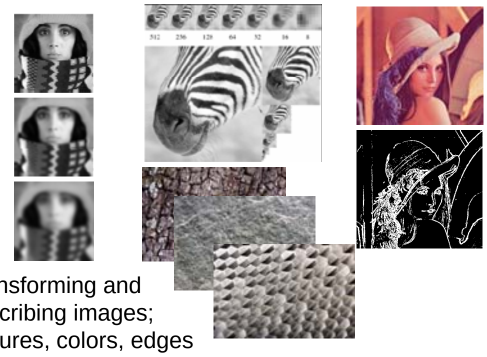
Epipolar geometry & stereo vision

Tuesday, Oct 21

Kristen Grauman
UT-Austin

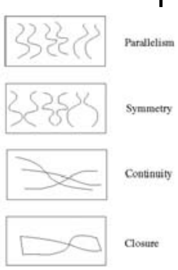


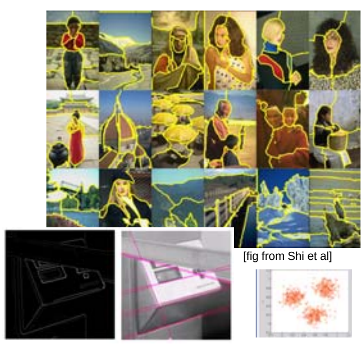
Recap: Features and filters



Transforming and describing images; textures, colors, edges

Recap: Grouping & fitting

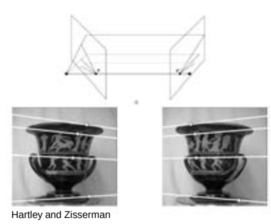


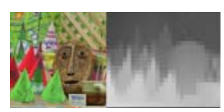


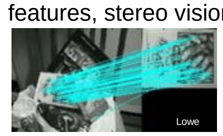
Clustering, segmentation, fitting; what parts belong together?

Now: Multiple views

Multi-view geometry, matching, invariant features, stereo vision







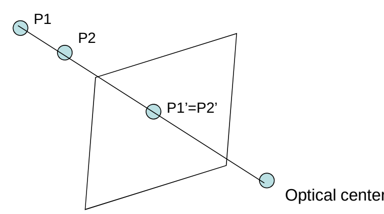
Why multiple views?

- Structure and depth are inherently ambiguous from single views.

Images from Lana Lazebnik

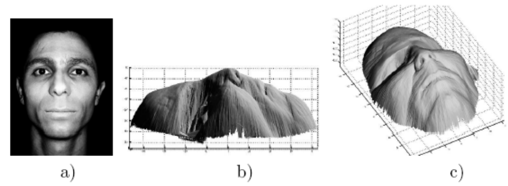
Why multiple views?

- Structure and depth are inherently ambiguous from single views.



- What cues help us to perceive 3d shape and depth?

Shading

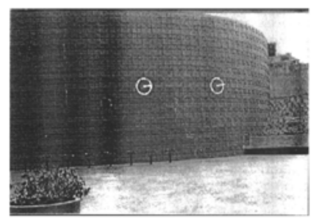


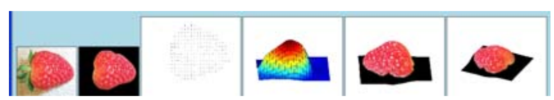
[Figure from Prados & Faugeras 2006]

Focus/Defocus

[Figure from H. Jin and P. Favaro, 2002]

Texture





[From A.M. Loh. The recovery of 3-D structure using visual texture patterns. PhD thesis]

Perspective effects

Image credit: S. Seitz

Motion

Figures from L. Zhang

http://www.brainconnection.com/teasers/?main=illusion/motion-shape

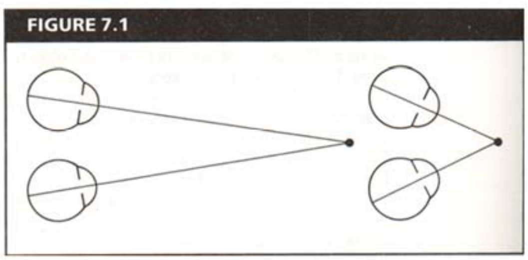
Estimating scene shape

- Shape from X: Shading, Texture, Focus, Motion…
- Stereo:
  – shape from “motion” between two views
  – infer 3d shape of scene from two (multiple) images from different viewpoints

Today

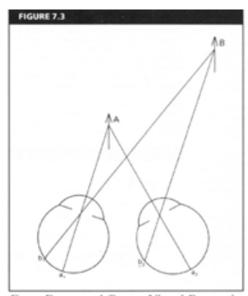
- Human stereopsis
- Stereograms
- Epipolar geometry and the epipolar constraint
  – Case example with parallel optical axes
  – General case with calibrated cameras
- Stereopsis
  – Finding correspondences along the epipolar line

Fixation, convergence



From Bruce and Green, Visual Perception, Physiology, Psychology and Ecology

Human stereopsis: disparity

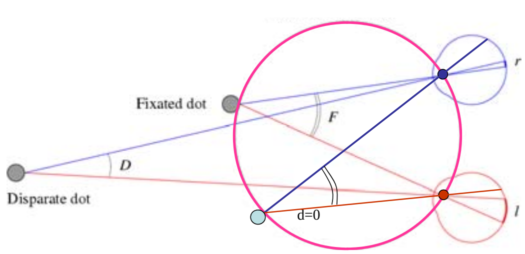


From Bruce and Green, Visual Perception, Physiology, Psychology and Ecology

Disparity occurs when eyes fixate on one object; others appear at different visual angles

Adapted from David Forsyth, UC Berkeley

Human stereopsis: disparity



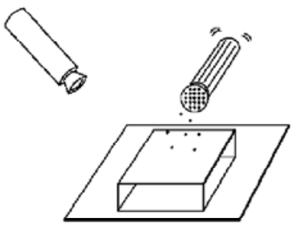
Disparity: $d = r\text{-}l = D\text{-}F$.

Adapted from M. Pollefeys

Random dot stereograms

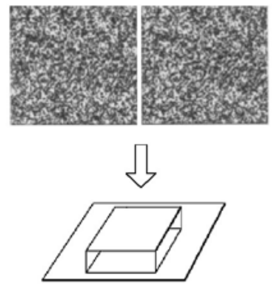
- Julesz 1960: Do we identify local brightness patterns before fusion (monocular process) or after (binocular)?
- To test: pair of synthetic images obtained by randomly spraying black dots on white objects

Random dot stereograms



Forsyth & Ponce

Random dot stereograms



Random dot stereograms

Figure 5.3.8 A random dot stereogram. These two images are derived from a single array of randomly placed squares by laterally displacing a region of them as described in the text. When they are viewed with crossed disparity (by crossing the eyes) so that the right eye's view of the left image is combined with the left eye's view of the right image, a square will be perceived to float above the page. (See pages 210–211 for instructions on fusing stereograms.)

From Palmer, "Vision Science", MIT Press

Random dot stereograms

- When viewed monocularly, they appear random; when viewed stereoscopically, see 3d structure.
- Conclusion: human binocular fusion not directly associated with the physical retinas; must involve the central nervous system
- Imaginary "*cyclopean retina"* that combines the left and right image stimuli as a single unit

Autostereograms

Exploit disparity as depth cue using single image

(Single image random dot stereogram, Single image stereogram)

Images from magiceye.com

Autostereograms

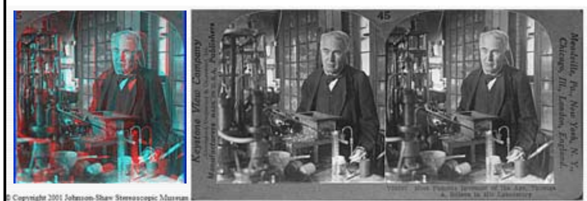
Images from magiceye.com

Stereo photography and stereo viewers

Take two pictures of the same subject from two slightly different viewpoints and display so that each eye sees only one of the images.

Invented by Sir Charles Wheatstone, 1838

Image courtesy of fisher-price.com



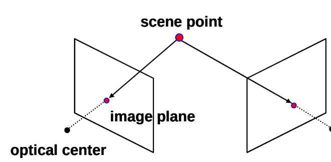
http://www.johnsonshawmuseum.org

http://www.johnsonshawmuseum.org

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

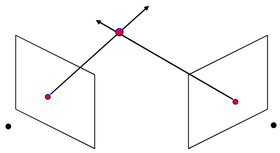
http://www.well.com/~jimg/stereo/stereo_list.html

Depth with stereo: basic idea



Source: Steve Seitz

Depth with stereo: basic idea

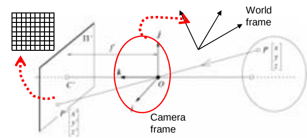


Basic Principle: Triangulation

- Gives reconstruction as intersection of two rays
- Requires
  - camera pose (calibration)
  - ***point correspondence***

Source: Steve Seitz

Camera calibration



Extrinsic parameters:
Camera frame ←→ Reference frame

Intrinsic parameters:
Image coordinates relative to camera ←→ Pixel coordinates

- *Extrinsic* params: rotation matrix and translation vector
- *Intrinsic* params: focal length, pixel sizes (mm), image center point, radial distortion parameters

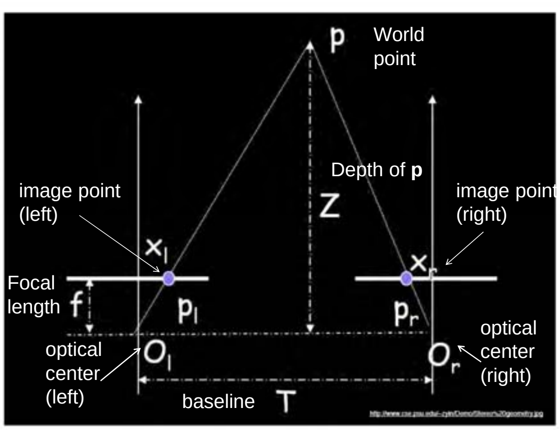
We'll assume for now that these parameters are given and fixed.

Today

- Human stereopsis
- Stereograms
- Epipolar geometry and the epipolar constraint
  - Case example with parallel optical axes
  - General case with calibrated cameras
- Stereopsis
  - Finding correspondences along the epipolar line

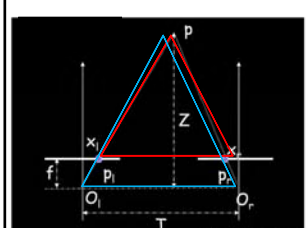
Geometry for a simple stereo system

- First, assuming parallel optical axes, known camera parameters (i.e., calibrated cameras):



Geometry for a simple stereo system

- Assume parallel optical axes, known camera parameters (i.e., calibrated cameras). We can triangulate via:



Similar triangles (p_l, P, p_r) and (O_l, P, O_r):

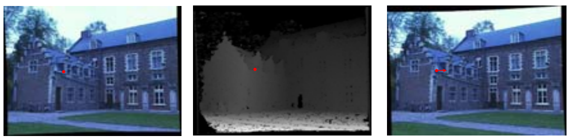
$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z}$$

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

disparity

Depth from disparity

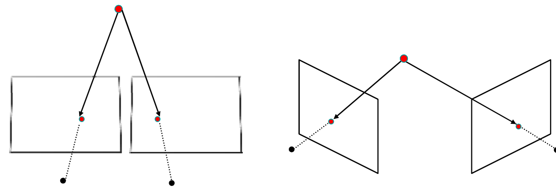
image I(x,y) | Disparity map D(x,y) | image I´(x´,y´)



(x´,y´)=(x+D(x,y), y)

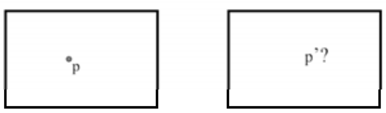
General case, with calibrated cameras

- The two cameras need not have parallel optical axes.



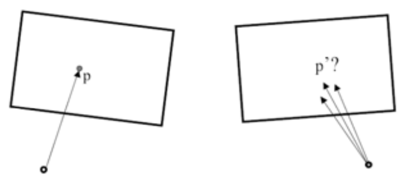
Vs.

Stereo correspondence constraints



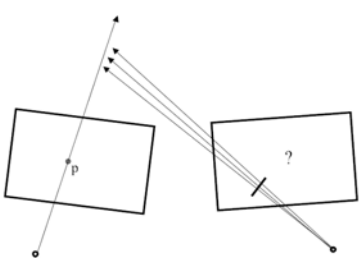
- Given p in left image, where can corresponding point p' be?

Stereo correspondence constraints



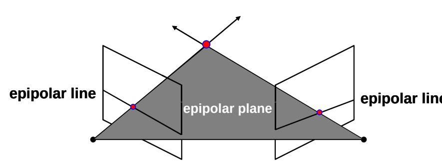
- Given p in left image, where can corresponding point p' be?

Stereo correspondence constraints



Stereo correspondence constraints

Geometry of two views allows us to constrain where the corresponding pixel for some image point in the first view must occur in the second view.

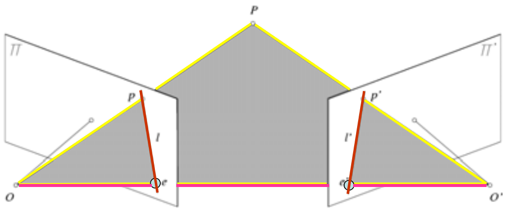


Epipolar constraint: Why is this useful?

- Reduces correspondence problem to 1D search along *conjugate epipolar lines*

Adapted from Steve Seitz

Epipolar geometry



• Epipolar Plane • Baseline

• Epipoles • Epipolar Lines

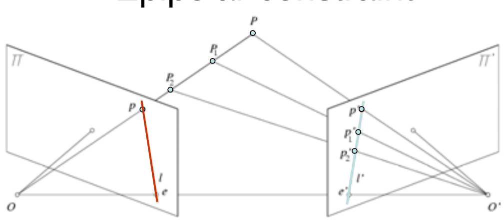
Adapted from M. Pollefeys, UNC

Epipolar geometry: terms

- **Baseline**: line joining the camera centers
- **Epipole**: point of intersection of baseline with the image plane
- **Epipolar plane**: plane containing baseline and world point
- **Epipolar line**: intersection of epipolar plane with the image plane

- All epipolar lines intersect at the epipole
- An epipolar plane intersects the left and right image planes in epipolar lines

Epipolar constraint

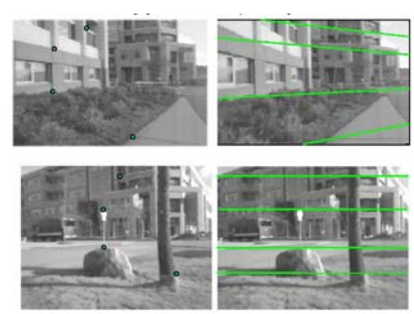


- Potential matches for *p* have to lie on the corresponding epipolar line *l'*.
- Potential matches for *p'* have to lie on the corresponding epipolar line *l*.

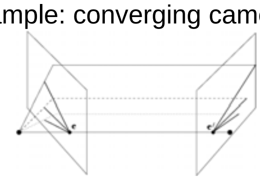
http://www.ai.sri.com/~luong/research/Meta3DViewer/EpipolarGeo.html

Source: M. Pollefeys

Example



Example: converging cameras



As position of 3d point varies, epipolar lines "rotate" about the baseline

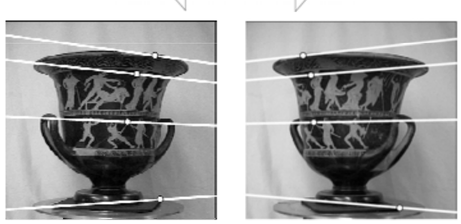


Figure from Hartley & Zisserman

Example: motion parallel with image plane

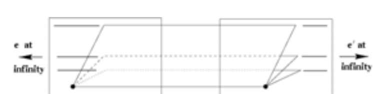
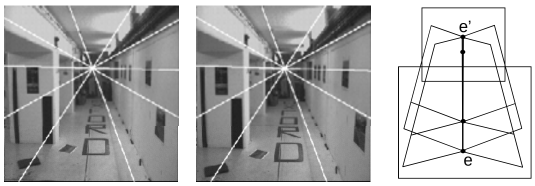


Figure from Hartley & Zisserman

Example: forward motion

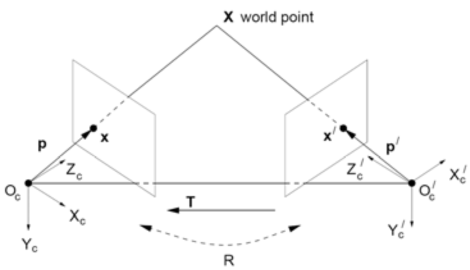


Epipole has same coordinates in both images.
Points move along lines radiating from e: “Focus of expansion”

Figure from Hartley & Zisserman

- For a given stereo rig, how do we express the epipolar constraints algebraically?

Stereo geometry, with calibrated cameras



If the rig is calibrated, we know :
how to **rotate** and **translate** camera reference frame 1 to get to camera reference frame 2.
Rotation: 3 x 3 matrix; translation: 3 vector.

Rotation matrix

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Express 3d rotation as series of rotations around coordinate axes by angles α, β, γ

Overall rotation is product of these elementary rotations:

$$\mathbf{R} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z$$

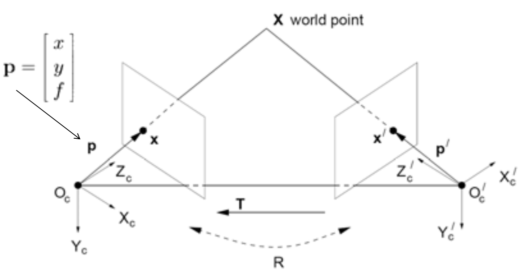
3d rigid transformation

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

$$\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T}$$

Stereo geometry, with calibrated cameras

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ f \end{bmatrix}$$



Camera-centered coordinate systems are related by known rotation **R** and translation **T**: $\mathbf{X}' = \mathbf{R}\mathbf{X} + \mathbf{T}$

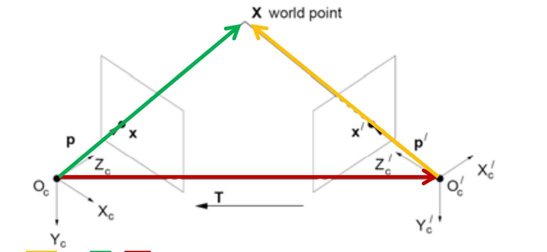
Cross product

$$\vec{a} \times \vec{b} = \vec{c} \qquad \begin{aligned} \vec{a} \cdot \vec{c} &= 0 \\ \vec{b} \cdot \vec{c} &= 0 \end{aligned}$$

Vector cross product takes two vectors and returns a third vector that's perpendicular to both inputs.

So here, c is perpendicular to both a and b, which means the dot product = 0.

From geometry to algebra



$$\boxed{\mathbf{X}'} = \mathbf{R}\boxed{\mathbf{X}} + \boxed{\mathbf{T}}$$

$$\underbrace{\mathbf{T} \times \mathbf{X}'}_{\text{Normal to the plane}} = \mathbf{T} \times \mathbf{RX} + \mathbf{T} \times \mathbf{T}$$

$$= \mathbf{T} \times \mathbf{RX}$$

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{X}') = \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{RX})$$

$$= 0$$

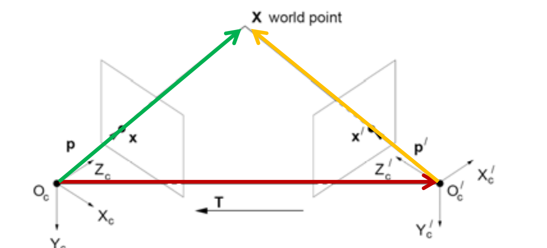
Matrix form of cross product

$$\vec{a} \times \vec{b} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = \vec{c} \qquad \begin{aligned} \vec{a} \cdot \vec{c} &= 0 \\ \vec{b} \cdot \vec{c} &= 0 \end{aligned}$$

Can be expressed as a matrix multiplication.

$$[a_x] = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \qquad \boxed{\vec{a} \times \vec{b} = [a_x]\vec{b}}$$

From geometry to algebra



$$\boxed{\mathbf{X}'} = \mathbf{R}\boxed{\mathbf{X}} + \boxed{\mathbf{T}}$$

$$\underbrace{\mathbf{T} \times \mathbf{X}'}_{\text{Normal to the plane}} = \mathbf{T} \times \mathbf{RX} + \mathbf{T} \times \mathbf{T}$$

$$= \mathbf{T} \times \mathbf{RX}$$

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{X}') \boxed{= \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{RX}) \\ = 0}$$

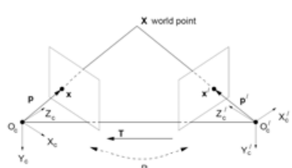
Essential matrix

$$\boxed{\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{RX}) = 0}$$

$$\mathbf{X}' \cdot (\mathbf{T}_x \ \mathbf{RX}) = 0$$

Let $\mathbf{E} = \mathbf{T}_x\mathbf{R}$

$$\mathbf{X}'^T \mathbf{E} \mathbf{X} = 0$$



This holds for the rays **p** and **p'** that are parallel to the camera-centered position vectors **X** and **X'**, so we have:

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

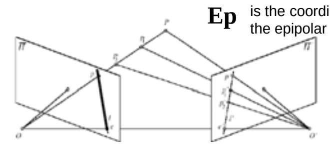
E is called the **essential matrix**, which relates corresponding image points [Longuet-Higgins 1981]

Essential matrix and epipolar lines

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

Epipolar constraint: if we observe point **p** in one image, then its position **p'** in second image must satisfy this equation.

$\mathbf{Ep}$ is the coordinate vector representing the epipolar line associated with point p



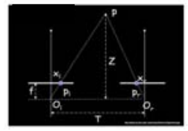
$\mathbf{E}^T\mathbf{p}'$ is the coordinate vector representing the epipolar line associated with point p'

Essential matrix: properties

- Relates image of corresponding points in both cameras, given rotation and translation
- Assuming intrinsic parameters are known

$$\mathbf{E} = \mathbf{T}_x\mathbf{R}$$

Essential matrix example: parallel cameras



$$\mathbf{R} = \mathbf{I}$$

$$\mathbf{T} = [-d, 0, 0]^T$$

$$\mathbf{E} = [\mathbf{T}_x]\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{pmatrix}$$

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

$$\begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & d \\ 0 & -d & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \end{bmatrix} = 0$$

$$\Leftrightarrow \begin{bmatrix} x' & y' & f \end{bmatrix} \begin{bmatrix} 0 \\ df \\ -dy \end{bmatrix} = 0$$

$$\boxed{\Leftrightarrow y = y'}$$

For the parallel cameras, image of any point must lie on same horizontal line in each image plane.

image I(x,y)

Disparity map D(x,y)

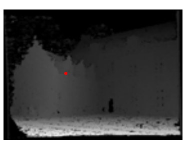


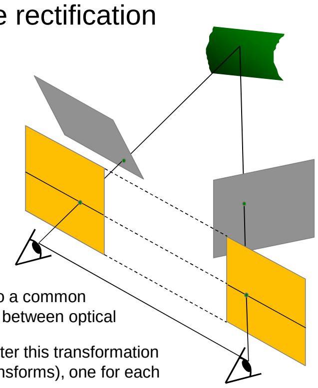
image I´(x´,y´)

(x´,y´)=(x+D(x,y),y)

What about when cameras' optical axes are not parallel?

Stereo image rectification

In practice, it is convenient if image scanlines are the epipolar lines.



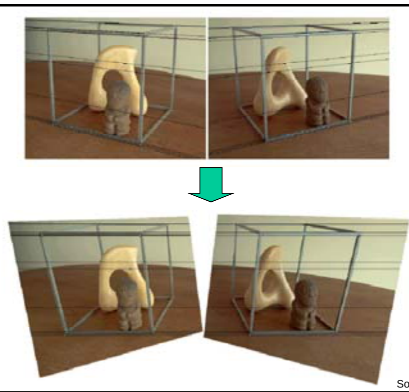
reproject image planes onto a common plane parallel to the line between optical centers

pixel motion is horizontal after this transformation

two homographies (3x3 transforms), one for each input image reprojection

Adapted from Li Zhang C. Loop and Z. Zhang. Computing Rectifying Homographies for Stereo Vision. CVPR 1999.

Stereo image rectification: example



Source: Alyosha Efros

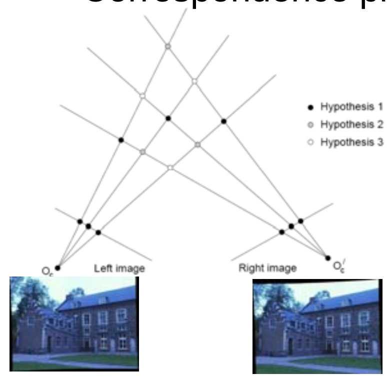
Today

- Human stereopsis
- Stereograms
- Epipolar geometry and the epipolar constraint
  - Case example with parallel optical axes
  - General case with calibrated cameras
- Stereopsis
  - Finding correspondences along the epipolar line

Stereo reconstruction: main steps

– Calibrate cameras
– Rectify images
– Compute disparity
– Estimate depth

Correspondence problem



Multiple match hypotheses satisfy epipolar constraint, but which is correct?

Figure from Gee & Cipolla 1999

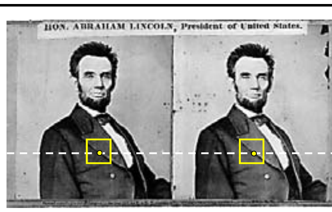
Correspondence problem

- To find matches in the image pair, we will assume
  – Most scene points visible from both views
  – Image regions for the matches are similar in appearance

Additional correspondence constraints

- Similarity
- Uniqueness
- Ordering
- Disparity gradient

Dense correspondence search

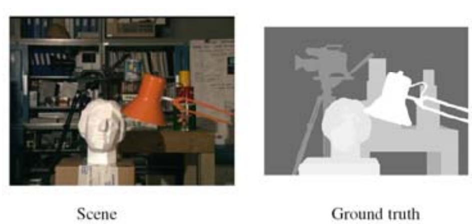


For each epipolar line
  For each pixel / window in the left image
  - compare with every pixel / window on same epipolar line in right image
  - pick position with minimum match cost (e.g., SSD, correlation)

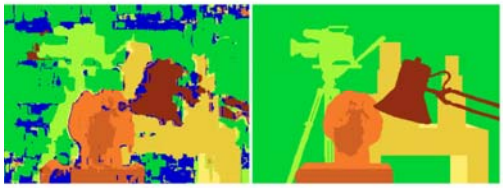
Adapted from Li Zhang

Example: window search

Data from University of Tsukuba



Example: window search



Window-based matching (best window size)

Ground truth

Effect of window size

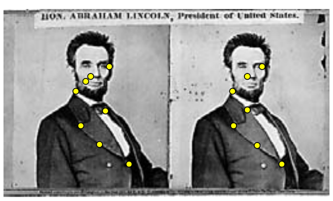
W = 3

W = 20

Want window large enough to have sufficient intensity variation, yet small enough to contain only pixels with about the same disparity.

Figures from Li Zhang

Sparse correspondence search



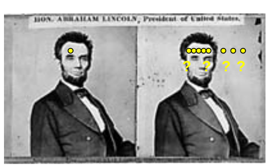
- Restrict search to sparse set of detected features
- Rather than pixel values (or lists of pixel values) use *feature descriptor* and an associated *feature distance*
- Still narrow search further by epipolar geometry

What would make good features?

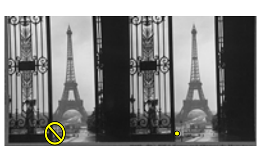
Dense vs. sparse

- Sparse
  - Efficiency
  - Can have more reliable feature matches, less sensitive to illumination than raw pixels
  - …But, have to know enough to pick good features; sparse info
- Dense
  - Simple process
  - More depth estimates, can be useful for surface reconstruction
  - …But, breaks down in textureless regions anyway, raw pixel distances can be brittle, not good with very different viewpoints

Difficulties in similarity constraint



Untextured surfaces



Occlusions

Uniqueness

- For opaque objects, up to one match in right image for every point in left image

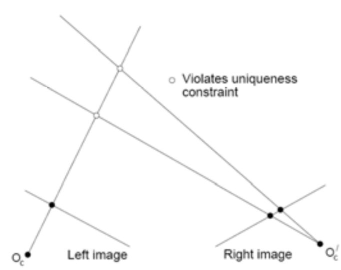


Figure from Gee & Cipolla 1999

Ordering

- Points on **same surface** (opaque object) will be in same order in both views

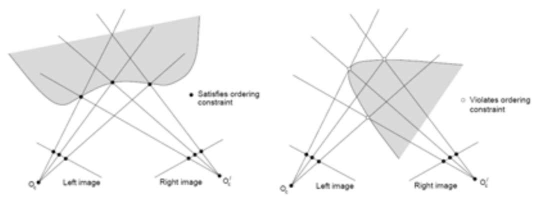
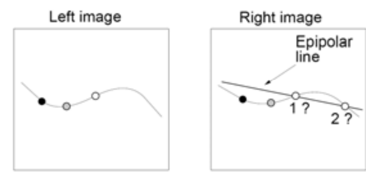


Figure from Gee & Cipolla 1999

Disparity gradient

- Assume piecewise continuous surface, so want disparity estimates to be locally smooth



Given matches ● and ⊙, point ○ in the left image must match point 1 in the right image. Point 2 would exceed the disparity gradient limit.

Figure from Gee & Cipolla 1999

Additional correspondence constraints

- Similarity
- Uniqueness
- Ordering
- Disparity gradient

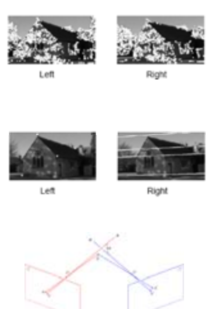
Epipolar lines constrain the search to a line, and these appearance and ordering constraints further reduce the possible matches.

Possible sources of error?

- Low-contrast / textureless image regions
- Occlusions
- Camera calibration errors
- Violations of *brightness constancy* (e.g., specular reflections)
- Large motions

Stereo reconstruction: main steps

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



Stereo in machine vision systems

Left : The Stanford cart sports a single camera moving in discrete increments along a straight line and providing multiple snapshots of outdoor scenes
Right : The INRIA mobile robot uses three cameras to map its environment

Forsyth & Ponce

Depth for segmentation

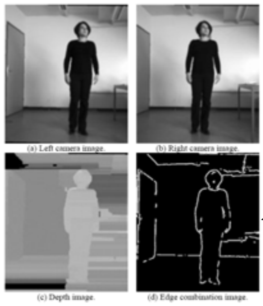


Figure 3 Stereo video frames with computed depth map and edge combination result.

Edges in disparity in conjunction with image edges enhances contours found

Danijela Markovic and Margrit Gelautz, Interactive Media Systems Group, Vienna University of Technology

Depth for segmentation

Danijela Markovic and Margrit Gelautz, Interactive Media Systems Group, Vienna University of Technology

Virtual viewpoint video

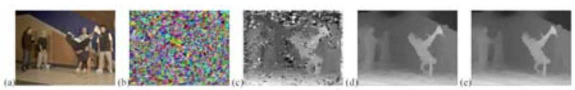


Figure 6: *Sample results from stereo reconstruction stage: (a) input color image; (b) color-based segmentation; (c) initial disparity estimates $d_{i,j}$; (d) refined disparity estimates; (e) smoothed disparity estimates $d_i(x)$.*

C. Zitnick et al, High-quality video view interpolation using a layered representation, SIGGRAPH 2004.

Virtual viewpoint video

http://research.microsoft.com/IVM/VVV/

Next

- Uncalibrated cameras
- Robust fitting