Advanced Computer Vision

Cameras and images

What is an image?

- Up until now: a function – a 2D pattern of intensity values
- Today: a 2D projection of 3D points

First Known Photograph – Heliograph

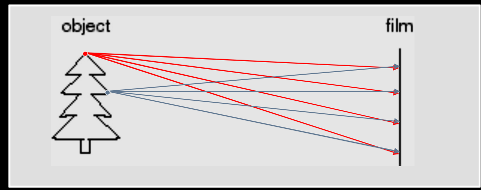
View from the Window at le Gras,
Joseph Nicéphore Niépce 1826

Reproduction, 1952

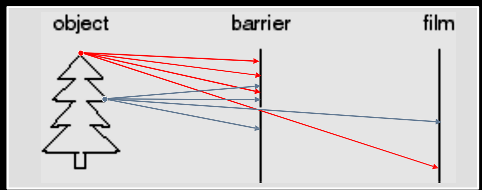
Projection

Projection

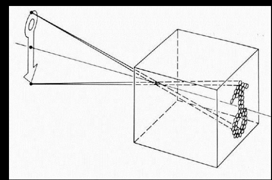
Image formation – (bad) method

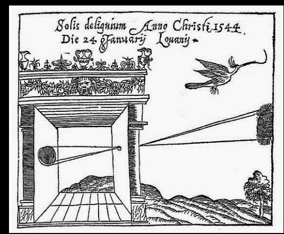


Pinhole camera



Camera Obscura (Latin: Darkened Room)



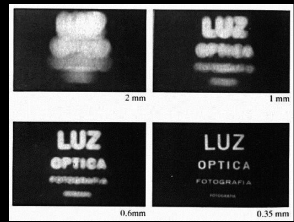


Home-made pinhole camera

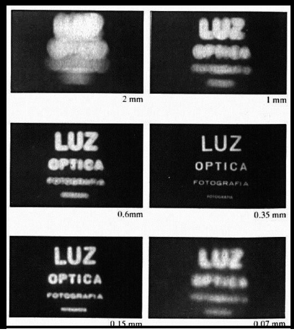
Why so blurry?

http://www.debevec.org/Pinhole/

Shrinking the aperture

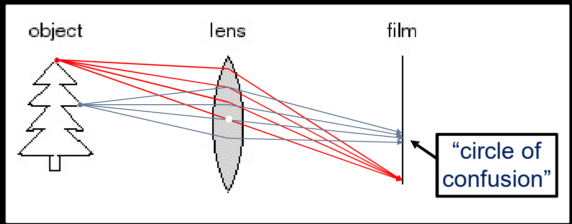


Shrinking the aperture

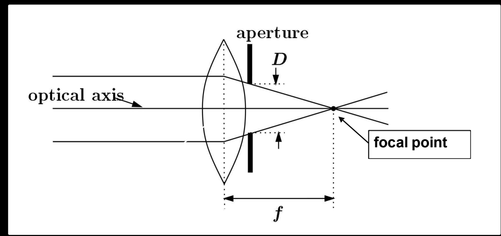


A little bit of computational photography

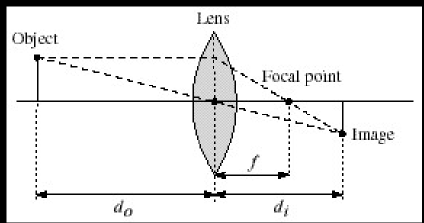
Adding a lens – and concept of focus



Lenses

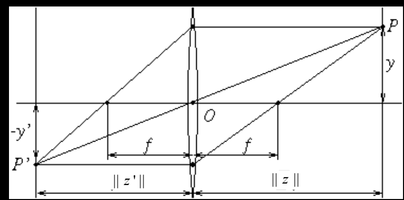


Thin lenses



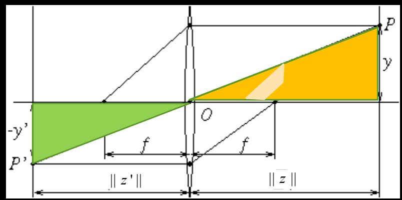
Slide by Steve Seitz

The thin lens



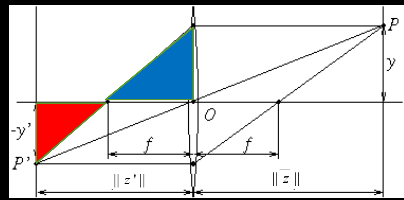
Computer Vision - A Modern Approach
Slides by D.A. Forsyth

The thin lens



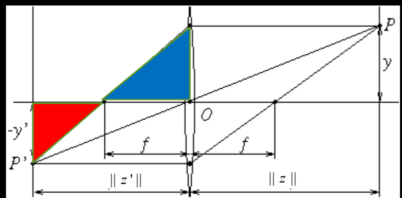
$$\frac{-y'}{y} = \frac{\| z' \|}{\| z \|}$$

The thin lens



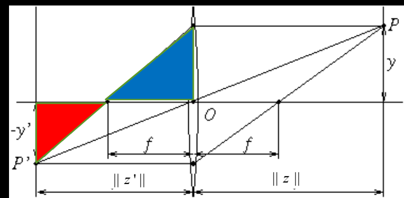
$$\frac{-y'}{y} = \frac{\| z' \|}{\| z \|} \qquad \frac{-y'}{y} = \frac{\| z' \| - f}{f} \qquad \rightarrow \frac{\| z' \|}{\| z \|} = \frac{\| z' \| - f}{f}$$

The thin lens equation



$$\frac{\| z' \|}{\| z \|} = \frac{\| z' \| - f}{f} \qquad \rightarrow \frac{1}{\| z \|} = \frac{1}{f} - \frac{1}{\| z' \|} \qquad \rightarrow \frac{1}{\| z' \|} + \frac{1}{\| z \|} = \frac{1}{f}$$

The thin lens equation



Any object point satisfying this equation is in focus.

$$\rightarrow \frac{1}{\| z' \|} + \frac{1}{\| z \|} = \frac{1}{f}$$

Thin lenses

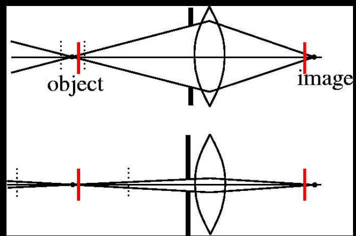
http://www.phy.ntnu.edu.tw/java/Lens/lens_e.html

(by Fu-Kwun Hwang)

Varying Focus

Ren Ng

Depth of field



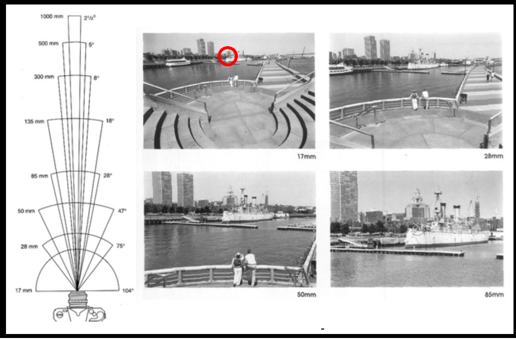
f / 5.6 = large aperture

f / 32 = small aperture

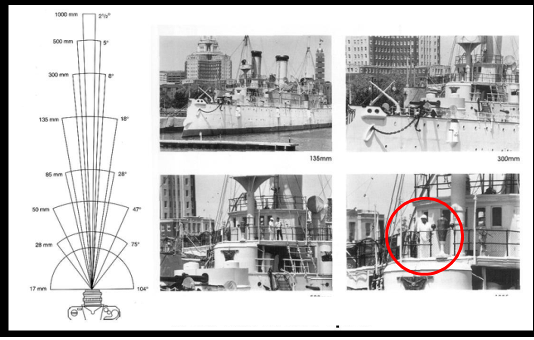
Flower images from Wikipedia http://en.wikipedia.org/wiki/Depth_of_field

Nice Depth of Field effect

Field of View (Zoom)



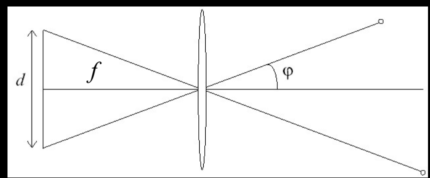
Field of View (Zoom)



FOV depends on Focal Length

d is the “retina” or sensor size

$$\phi = 2\tan^{-1}\left(\frac{d/2}{f}\right)$$



Larger Focal Length => Smaller FOV

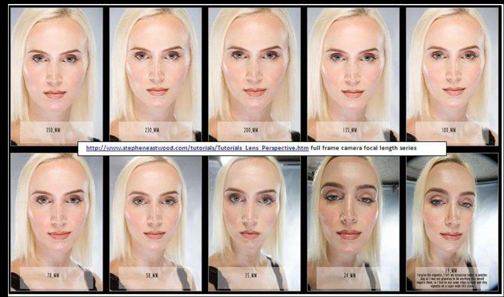
Zooming and Moving are not the same…

Field of View / Focal Length

Large FOV, small *f*
Camera close to car

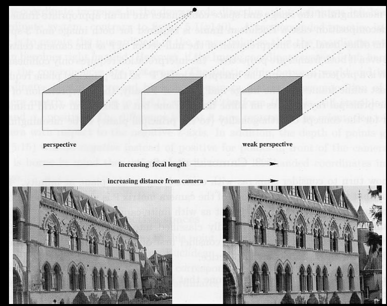
Small FOV, large *f*
Camera far from the car

Perspective and Portraits



Perspective and Portraits

Effect of focal length on perspective effect



From Zisserman & Hartley

Dolly Zoom

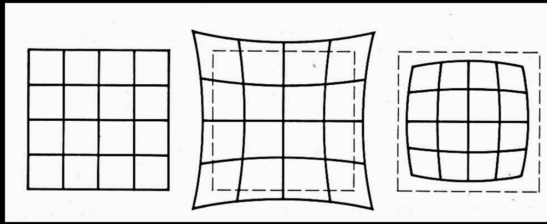

Pioneered by Hitchcock in *Vertigo* (1958)

Original(YouTube link) (2:07)   Widely used (YouTube link)

Jim Rehg

But reality can be a problem…

Geometric Distortion

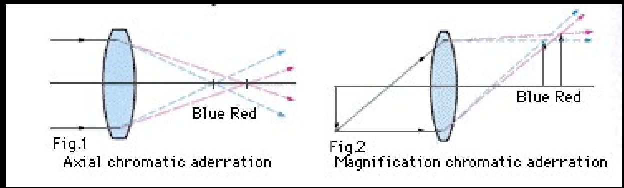


No distortion   Pin cushion   Barrel

Correcting radial distortion

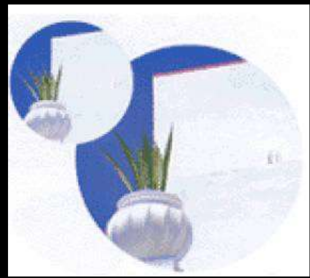

from Helmut Dersch

Chromatic Aberration

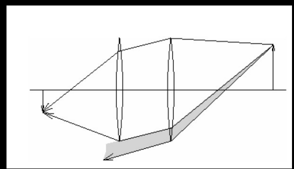


Rays of different wavelength
focus in different planes

Chromatic Aberration

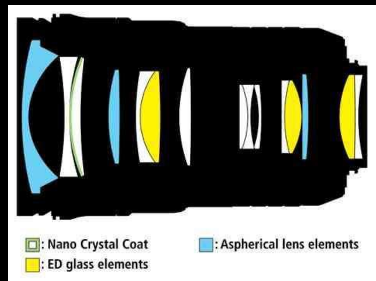


Vignetting



Lens systems

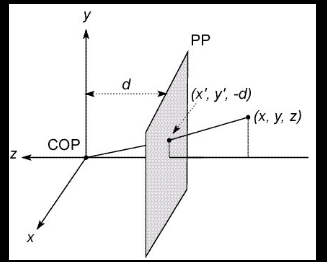
Nikon 24-70mm zoom



Perspective imaging

Modeling projection – coordinate system

- We will use the pin-hole model as an approximation
- Put the optical center (Center Of Projection) at the origin
- STANDARD (x,y) COORDINATE SYSTEM
- Put the image plane (Projection Plane) in front of the COP (why?)
- The camera looks down the ***negative*** z axis



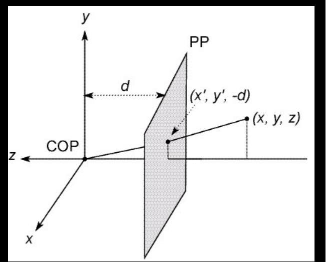
Modeling projection

Projection equations

- Compute intersection with Perspective Projection of ray from (x,y,z) to COP
- Derived using similar triangles

$$(X, Y, Z) \rightarrow (-d\frac{X}{Z}, -d\frac{Y}{Z}, -d)$$

(assumes normal Z negative – we'll change later)



Modeling projection

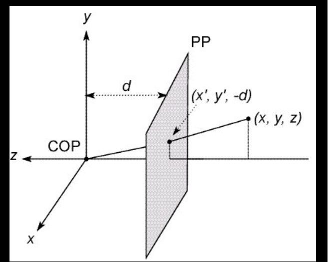
Distant objects are smaller

Projection equations

$$(X, Y, Z) \rightarrow (-d\frac{X}{Z}, -d\frac{Y}{Z}, -d)$$

We get the projection by throwing out the last coordinate:

$$(x', y') = (-d\frac{X}{Z}, -d\frac{Y}{Z})$$



Quiz

- When objects are very far away, the real X and real X can be huge. If I move the camera (the origin) those numbers hardly change. This explains:

a) Why the moon follows you.
b) Why the North Star is always North.
c) Why you can tell time from the Sun regardless of where you are?
d) All of the above.

Homogeneous coordinates

Is this a linear transformation?

No – division by the (not constant) Z is non-linear

Trick: add one more coordinate:

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous image (2D) coordinates

$$(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

homogeneous scene (3D) coordinates

Homogeneous coordinates

Converting *from* homogeneous coordinates:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w) \qquad \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

(this makes homogenous coordinates invariant under scale)

Perspective Projection

- Projection is a matrix multiply using homogeneous coordinates:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z/f \end{bmatrix} \Rightarrow \left(f\frac{x}{z}, f\frac{y}{z} \right)$$

$$\Rightarrow (u, v)$$

S. Seitz

Perspective Projection

- How does scaling the projection matrix change the transformation?

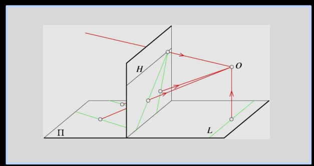
Perspective Projection

- How does scaling the projection matrix change the transformation?

$$\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} fx \\ fy \\ z \end{bmatrix} \Rightarrow \left(f\frac{x}{z}, f\frac{y}{z} \right)$$

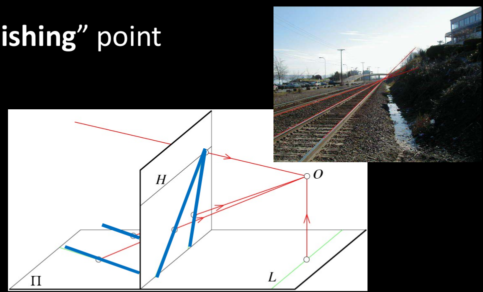
Geometric properties of projection

- Points go to **points**
- Lines go to **lines**
- Polygons go to **polygons**



Parallel lines in the world meet in the image

"**Vanishing**" point



Parallel lines converge in math too…

Line in 3-space

$$x(t) = x_0 + at$$
$$y(t) = y_0 + bt$$
$$z(t) = z_0 + ct$$

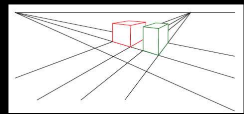
Perspective projection of the line

$$x'(t) = \frac{fx}{z} = \frac{f(x_0 + at)}{z_0 + ct}$$
$$y'(t) = \frac{fy}{z} = \frac{f(y_0 + bt)}{z_0 + ct}$$

In the limit as $t \to \pm\infty$ we have (for $c \neq 0$): $x'(t) \to \frac{fa}{c}, \quad y'(t) \to \frac{fb}{c}$

Vanishing points

- Each set of parallel lines (=direction) meets at a different point
- Sets of parallel lines on the same plane lead to collinear vanishing points.
  - The line is called the horizon for that plane

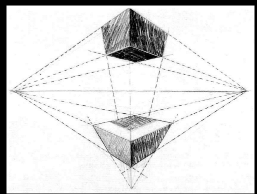


- Good ways to spot faked images
  - scale and perspective don't work
  - vanishing points behave badly

More vanishing points

3-point perspective: Different directions correspond to different vanishing points

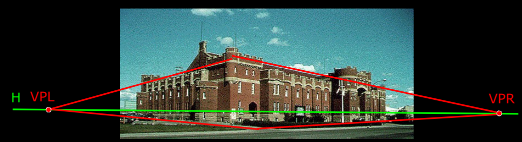
VP_1



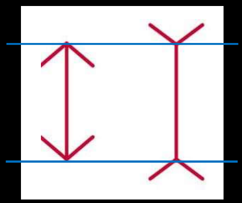
VP_2

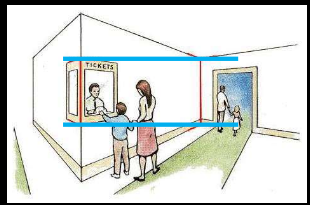
VP_3

Vanishing points



Human vision: Müller-Lyer Illusion





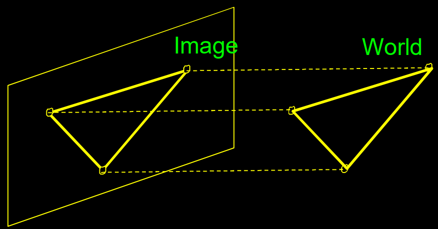
Which line is longer? http://www.michaelbach.de/ot/sze_muelue/index.html

Quiz

What determines at what point in the image parallel lines intersect?

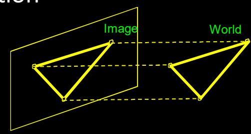
a) The direction the lines have in the world.
b) Whether the world lines are on the ground plane?
c) The orientation of the camera.
d) (a) and (c)

Other models: Orthographic projection



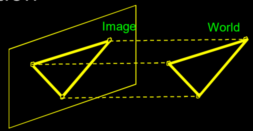
Other models: Orthographic projection

- Special case of perspective projection
  - Distance from the COP to the image plane is infinite
    => Both f and Z are very large
  - Good approximation for telephoto optics
  - Also called "parallel projection": (x, y, z) → (x, y)



Other models: Orthographic projection

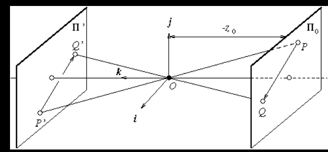
- Special case of perspective projection
  - What's the projection matrix?



$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \Rightarrow (x, y)$$

Other projection models: Weak perspective

- *Perspective* effects, but not over the scale of *individual* objects
- Collect points into a group at about the same depth, then divide each point by the depth of its group
- Advantage: easy
- Disadvantage : only approximate



$$(x, y, z) \to \left(\frac{fx}{z_0}, \frac{fy}{z_0} \right)$$

Other projection models: Weak perspective

- *Perspective* effects, but not over the scale of *individual* objects
- Collect points into a group at about the same depth, then divide each point by the depth of its group
- Advantage: easy
- Disadvantage : only approximate

$$(x, y, z) \to \left(\frac{fx}{z_0}, \frac{fy}{z_0} \right)$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/s \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1/s \end{bmatrix} \Rightarrow (sx, sy)$$

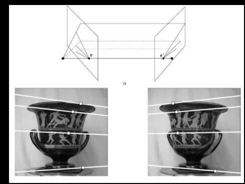
Three camera projections

	3-d point	2-d image position
(1) Perspective:	$(x, y, z) \to$	$\left(\frac{fx}{z}, \frac{fy}{z} \right)$
(2) Weak perspective:	$(x, y, z) \to$	$\left(\frac{fx}{z_0}, \frac{fy}{z_0} \right)$
(3) Orthographic:	$(x, y, z) \to$	(x, y)

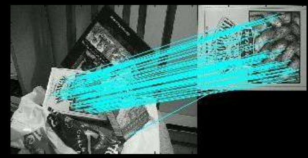
Computer Vision

Stereo geometry

Stereo: A Special case of Multiple views



Hartley and Zisserman



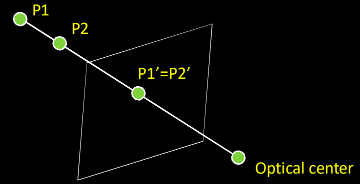
Why multiple views?

- Structure and depth are inherently ambiguous from single views.

Images from S. Lazebnik

Why multiple views?

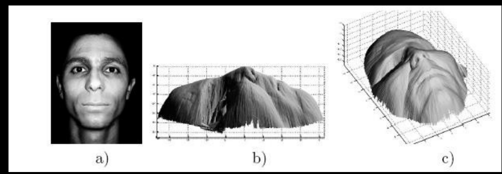
- Structure and depth are inherently ambiguous from single views.



Perspective effects

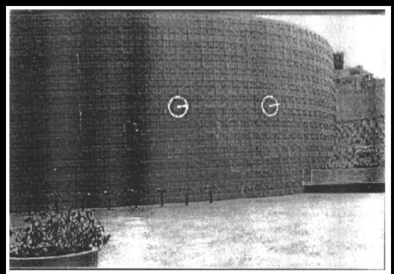
S. Seitz

Shading



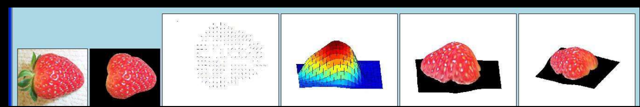
K. Grauman

Texture



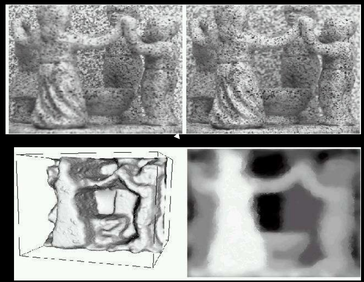
[A.M. Loh. The recovery of 3-D structure using visual texture patterns.]

Texture



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

Focus/defocus



Images from same point of view, different camera parameters

3d shape / depth estimates

Figures from H. Jin and P. Favaro, 2002

Motion

Figures from L. Zhang

Shape cues

Estimating scene shape from one eye

- "Shape from X": Shading, Texture, Focus, Motion…
- Very popular circa 1980

But we (and lots of creatures) have two eyes!

Stereo:

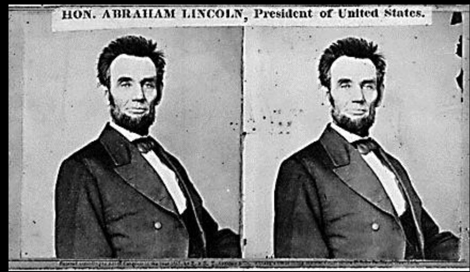
- The image from one eye is a little different than the image form he other eye.
- Think of shape from "motion" between two views
- Infer 3d shape of scene from two (multiple) images from different viewpoints

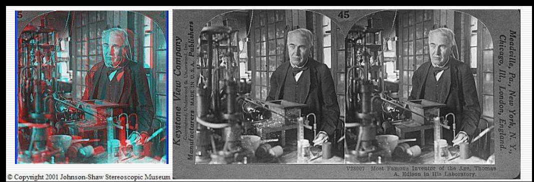
Stereo photography and stereo viewers

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

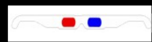
Invented by Sir Charles Wheatstone 1838

People fascinated by 3D





http://www.johnsonshawmuseum.org



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

Teesta suspension bridge
Darjeeling, India

Stereo photography and stereo viewers

When I grew up…

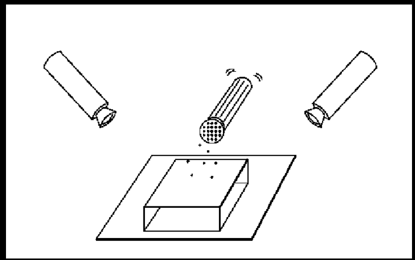
Stereo photography and stereo viewers

You guys..

The Basic Idea: Two slightly different images

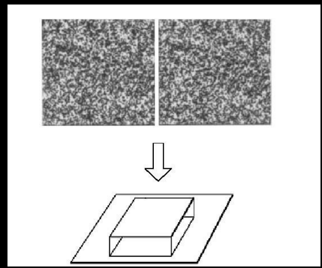
http://www.well.com/~jimg/stereo/stereo_list.html

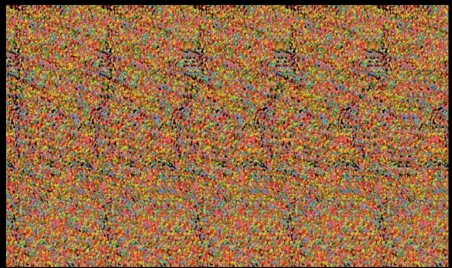
Random dot stereograms



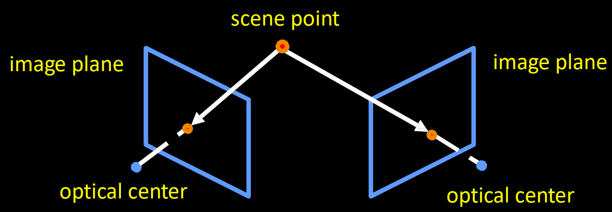
Forsyth & Ponce

Random dot stereograms





Basic stereo geometry

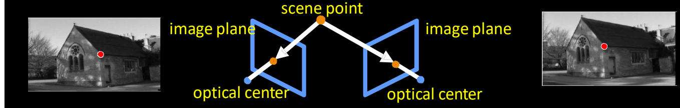


Estimating depth with stereo

Stereo: shape from "motion" between two views

We'll need to consider:

- Info on camera pose ("calibration")
- Image point correspondences



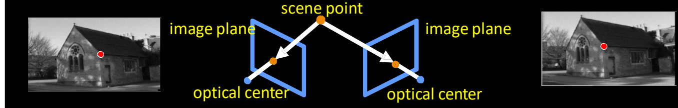
Estimating depth with stereo

Stereo: shape from "motion" between two views

We'll need to consider:

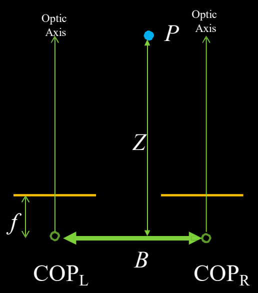
Info on camera pose ("calibration")

Image point correspondences



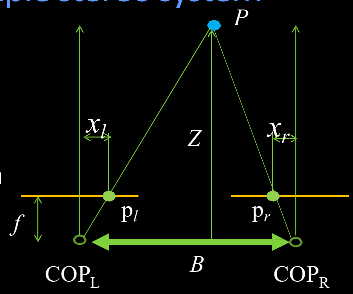
Geometry for a simple stereo system

- First, assuming parallel optical axes, known camera parameters (i.e., calibrated cameras)
- Figure is looking down on the cameras and image planes
- Baseline B, focal length f
- Point P is distance Z in camera coordinate systems



Geometry for a simple stereo system

- Point P projects into left and right images.
- Distance is positive in left image, and negative in right



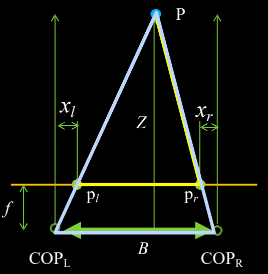
Geometry for a simple stereo system

- What is the expression for Z?
- Similar triangles (p_l, P, p_r) and (C_l, P, C_r):

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

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

Disparity



Depth from disparity

image I(x,y)

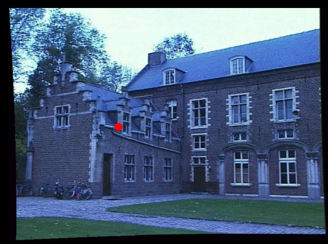


image I'(x,y)

Depth from disparity

image I(x,y)

Disparity map D(x,y)

Depth from disparity

(x´,y´)=(x+D(x,y), y)

image I´(x´,y´)

Computer Vision

Epipolar geometry

Depth from disparity

image I(x,y)

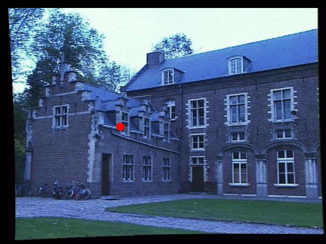


image I'(x,y)

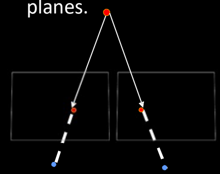
Depth from disparity

image I(x,y)

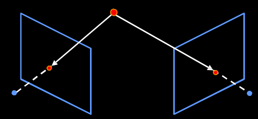
Disparity map D(x,y)

General case, with calibrated cameras

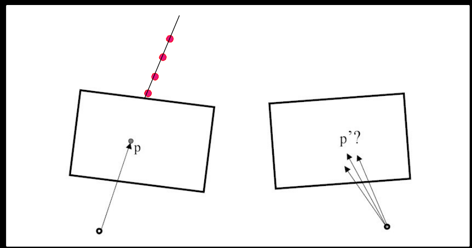
- The two cameras need not have parallel optical axes and image planes.



vs.



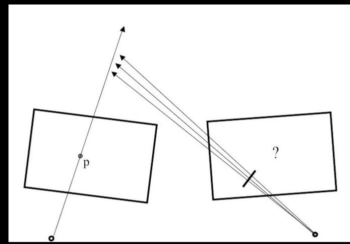
Stereo correspondence constraints



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

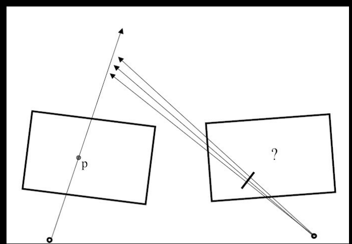
Stereo correspondence constraints

Remember: in perspective projection, lines project into lines.

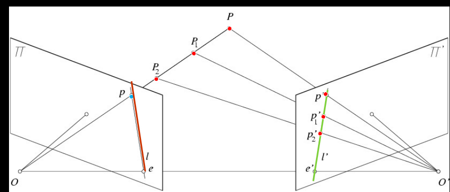


Stereo correspondence constraints

So the ***line*** containing the center of projection and the point P in the left image must project to a ***line*** in the right image.



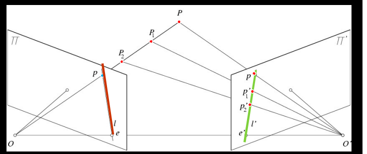
Epipolar constraint



Geometry of two views constrains where the corresponding pixel for some image point in the first view must occur in the second view.

Epipolar geometry: Terms

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



Why is the epipolar constraint useful?

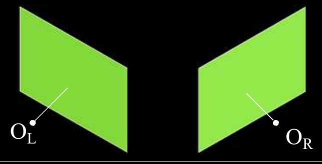
Epipolar constraint

The *epipolar constraint* reduces the correspondence problem to a 1D search along an epipolar line.

Image from Andrew Zisserman

What do the epipolar lines look like?

1.



2.

Example: converging cameras

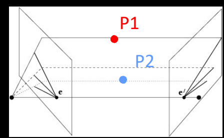
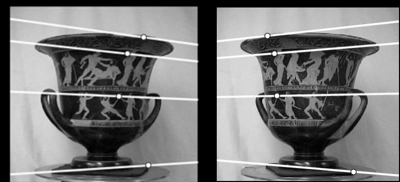
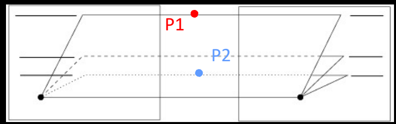


Figure from Hartley & Zisserman



Example: parallel image planes



Where are the epipoles?

Quiz: two stereo pairs

a)

b)

Quiz:

How do we know that (B) has parallel image planes

a) The epipolar lines are horizontal
b) The epipolar lines are parallel
c) Because I just said (B) had parallel iimage planes

Computer Vision

Stereo correspondence

For now assume parallel image planes

- Assume parallel (co-planar) image planes…
- Assume same focal lengths…
- Assume epipolar lines are horizontal…
- Assume epipolar lines are at the same y location in the image…

- That's a lot of assuming, but it allows us to move to the correspondence problem – which you will be solving!

Correspondence problem

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

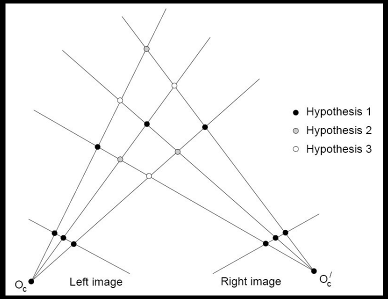


Figure from Gee & Cipolla 1999

Correspondence problem

Beyond the hard constraint of epipolar geometry, there are "soft" constraints to help identify corresponding points

- Similarity
- Uniqueness
- Ordering
- Disparity gradient is limited

Correspondence problem

Beyond the hard constraint of epipolar geometry, there are "soft" constraints to help identify corresponding points

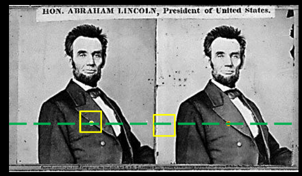
- **Similarity**
- Uniqueness
- Ordering
- Disparity gradient is limited

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

Dense correspondence search

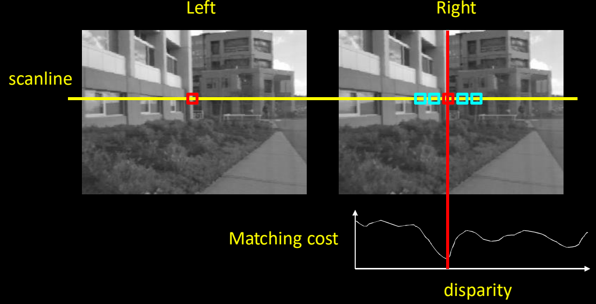


For each pixel / window in the left image

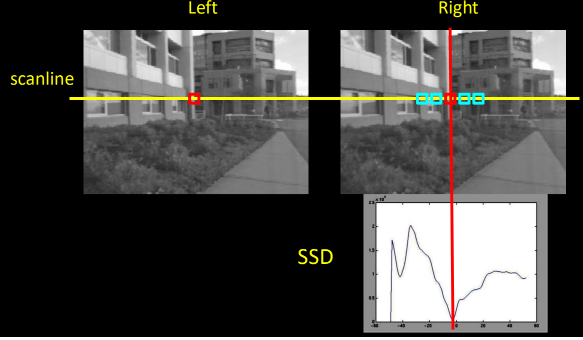
- Compare with every pixel / window on same epipolar line in right image
- Pick position with minimum match cost (e.g., SSD, normalized correlation)

Adapted from Li Zhang

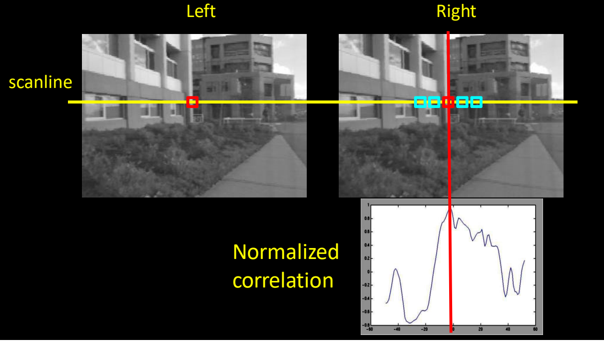
Correspondence search: (dis)similarity constraint



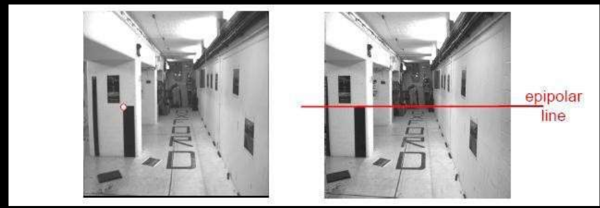
Correspondence search: (dis)similarity constraint



Correspondence search: similarity constraint

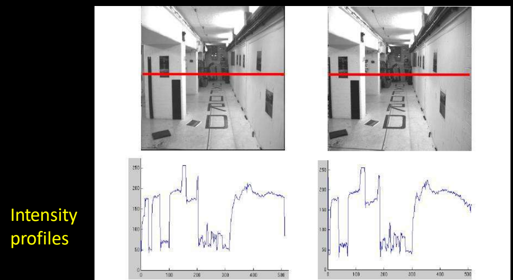


Correspondence problem



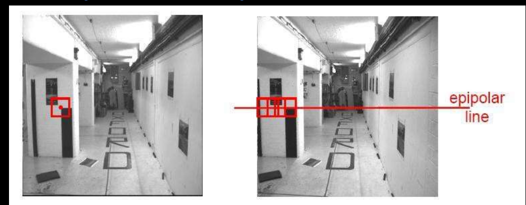
Source: Andrew Zisserman

Correspondence problem



Source: Andrew Zisserman

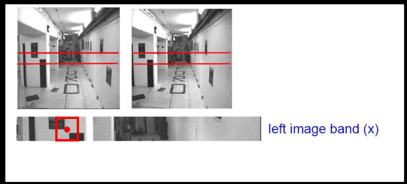
Correspondence problem



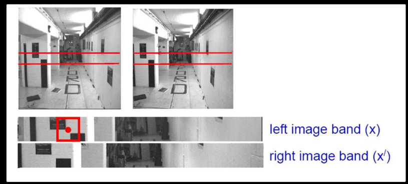
Neighborhoods of corresponding points
are similar in intensity patterns.

Source: Andrew Zisserman

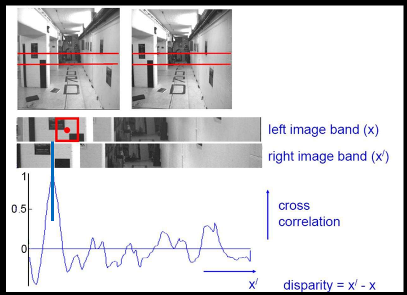
Correlation-based window matching



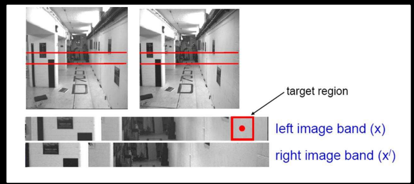
Correlation-based window matching



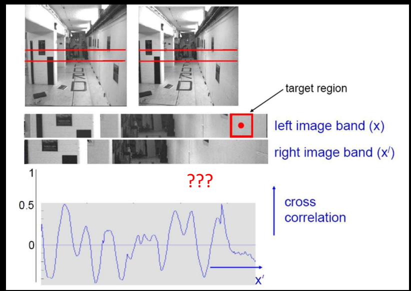
Correlation-based window matching



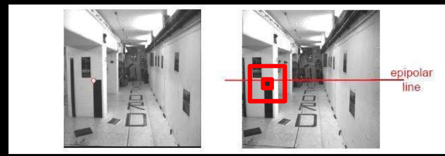
Correlation-based window matching



Correlation-based window matching



Effect of window size



Source: Andrew Zisserman

Effect of window size

W = 3 W = 20

Figures from Li Zhang

Correspondence problem

Beyond the hard constraint of epipolar geometry, there are “soft” constraints to help identify corresponding points

- Similarity
- Uniqueness
- Ordering
- Disparity gradient is limited

Uniqueness constraint

No more than one match in right image for every point in left image

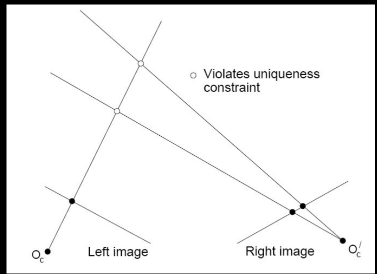
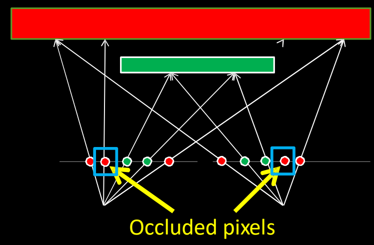


Figure from Gee & Cipolla 1999

Problem: Occlusion



Ordering constraint

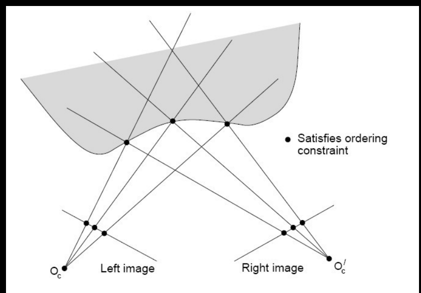
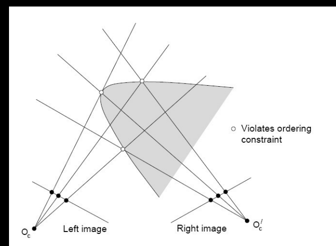


Figure from Gee & Cipolla 1999

Ordering constraint

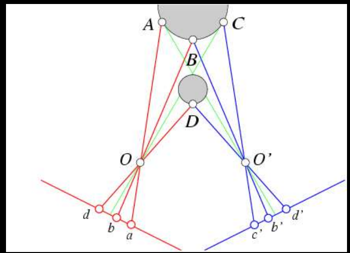
Won't always hold, e.g. consider transparent object…



Figures from Forsyth & Ponce

Ordering constraint

…or a narrow occluding surface

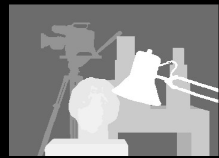


Figures from Forsyth & Ponce

Stereo results

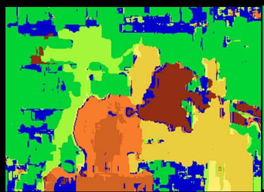
Image data from University of Tsukuba

Scene



Ground truth

Results with window search



Window-based matching
(best window size)

Ground truth

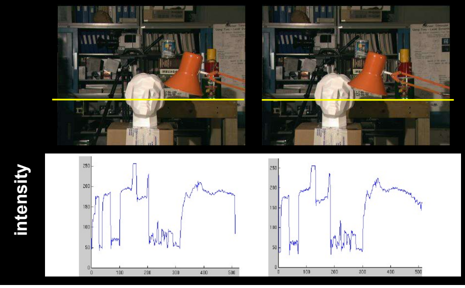
Better solutions

Beyond individual correspondences to estimate disparities:

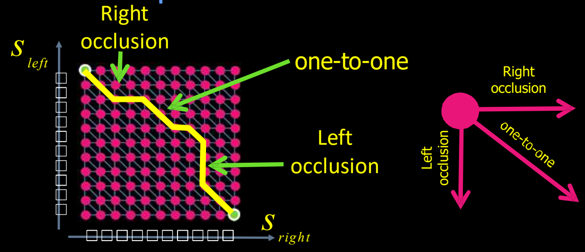
- Optimize correspondence assignments jointly
  - Scanline at a time (DP)
  - Full 2D grid (graph cuts)

Scanline stereo

Coherently match pixels on the entire scanline



"Shortest paths" for scan-line stereo



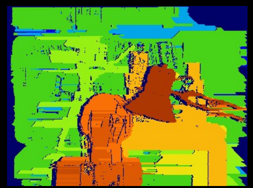
Can be implemented with dynamic programming

Ohta & Kanade '85, Cox et al. '96, Intille & Bobick, '01

Slide: Y. Boykov

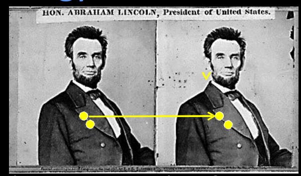
Coherent stereo on 2D grid

Scanline stereo generates streaking artifacts



Can't use dynamic programming to find spatially coherent disparities/ correspondences on a 2D grid

Stereo as energy minimization



What defines a good stereo correspondence?

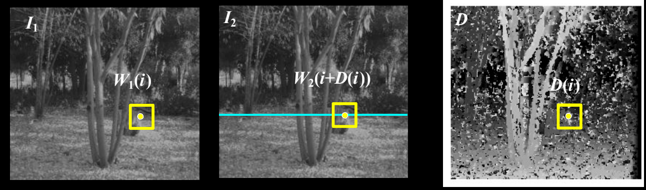
1. ***Match quality*** - Want each pixel to find a good appearance match in the other image
2. ***Smoothness*** - of two pixels are adjacent, they should (usually) move about the same amount

Stereo matching as energy minimization

Data term: $E_{data} = \sum_i \left(W_1(i) - W_2(i + D(i))\right)^2$

Source: Steve Seitz

Stereo matching as energy minimization



Smoothness term: $E_{smooth} = \sum_{\text{neighbors } i,j} \rho\left(D(i) - D(j)\right)$

Source: Steve Seitz

Stereo matching as energy minimization

Total energy: $E = \alpha E_{data}(I_1, I_2, D) + \beta E_{smooth}(D)$

Source: Steve Seitz

Better results…

- Energy functions of this form can be minimized using *graph cuts*
- Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 2001

Better results…

State of the art method

Ground truth

For the latest and greatest: http://www.middlebury.edu/stereo

Challenges

- Low-contrast ; textureless image regions
- Occlusions
- Violations of brightness constancy (e.g., specular reflections)
- Really large baselines (foreshortening and appearance change)
- Camera calibration errors