Scene Modeling for a Single View

Reading:

A. Criminisi, I. Reid and A. Zisserman, “Single View Metrology” (ICCV 99)

B. Zisser And Mundy, appendix

on to 3D…

We want real 3D scene walk-throughs:

- Camera rotation
- Camera translation

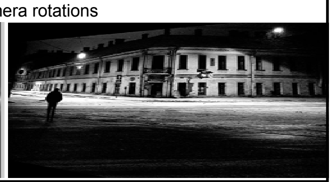
Can we do it from a single photograph?

Camera rotations with homographies

Original image

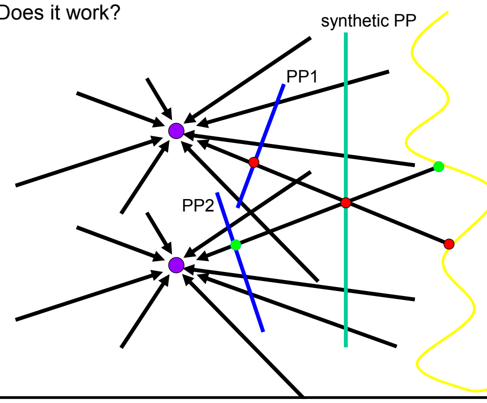
St.Petersburg
photo by A. Tikhonov

Virtual camera rotations

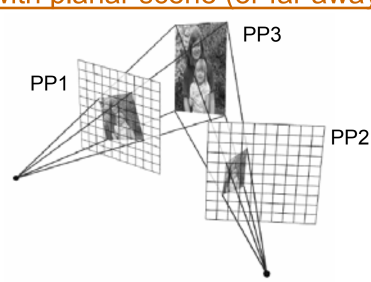


Camera translation

Does it work?



Yes, with planar scene (or far away)



PP3 is a projection plane of both centers of projection, so we are OK!

So, what can we do here?

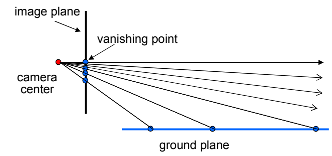
Model the scene as a set of planes!

Now, just need to find the orientations of these planes.

Silly Euclid: Trix are for kids!

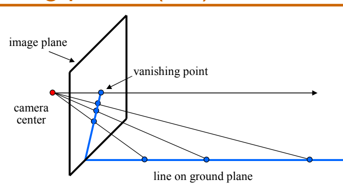
Parallel lines???

Vanishing points

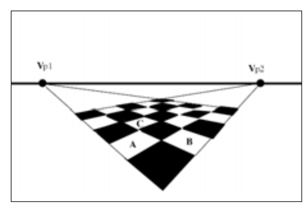


Vanishing point
- projection of a point at infinity
- Caused by ideal line

Vanishing points (2D)

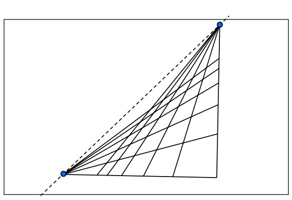


Vanishing lines



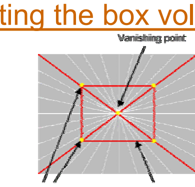
Multiple Vanishing Points
- Any set of parallel lines on the plane define a vanishing point
- The union of all of these vanishing points is the *horizon line*
  - also called *vanishing line*
- Note that different planes define different vanishing lines

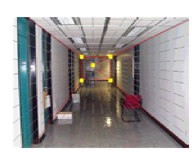
Vanishing lines



Multiple Vanishing Points
- Any set of parallel lines on the plane define a vanishing point
- The union of all of these vanishing points is the *horizon line*
  - also called *vanishing line*
- Note that different planes define different vanishing lines

Fitting the box volume

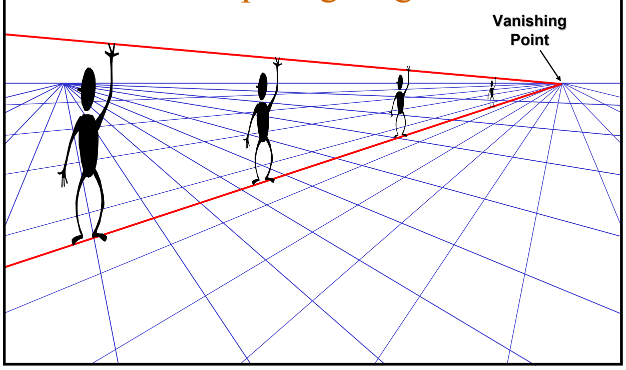




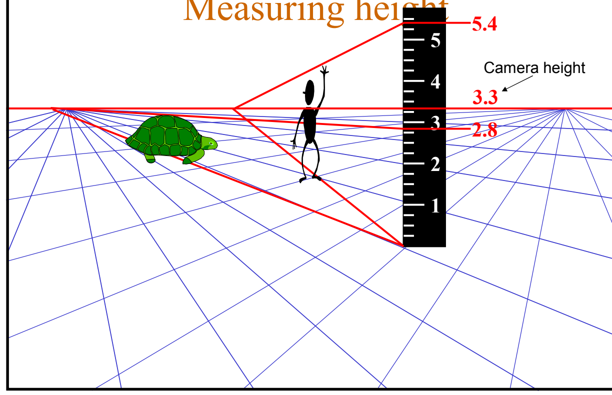
Q: What's the significance of the vanishing point location?

A: It's at eye level: ray from COP to VP is perpendicular to image plane.

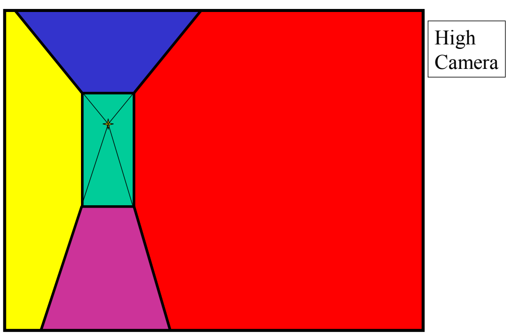
Comparing heights



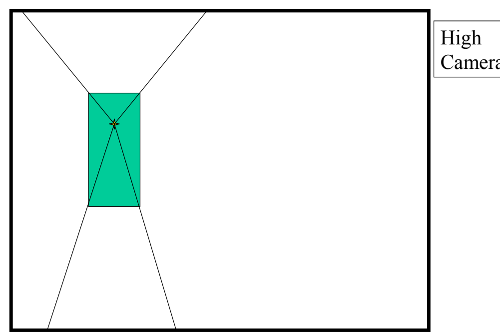
Measuring height



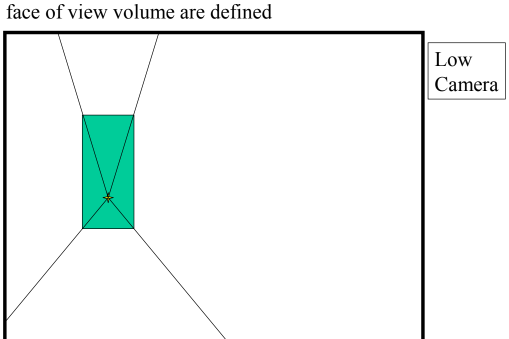
Example of user input: vanishing point and back face of view volume are defined



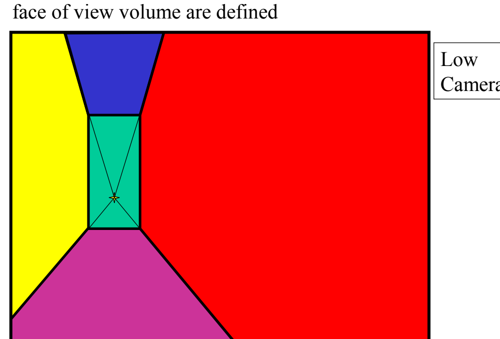
Example of user input: vanishing point and back face of view volume are defined



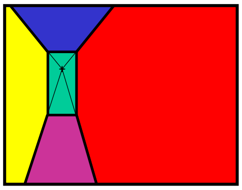
Example of user input: vanishing point and back face of view volume are defined



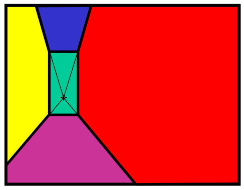
Example of user input: vanishing point and back face of view volume are defined



Comparison of how image is subdivided based on two different camera positions. You should see how moving the vanishing point corresponds to moving the eyepoint in the 3D world.

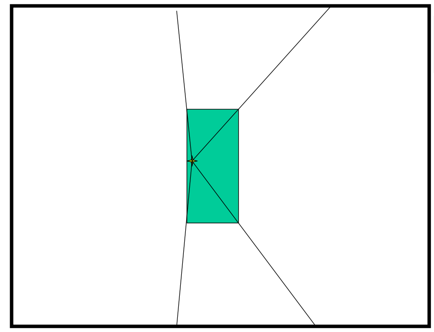


High Camera



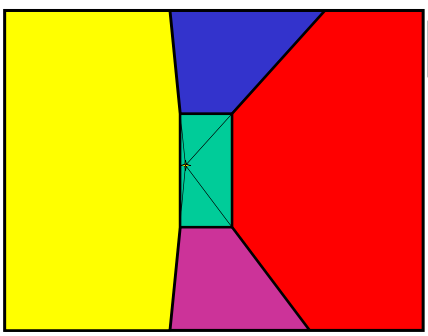
Low Camera

Another example of user input: vanishing point and back face of view volume are defined



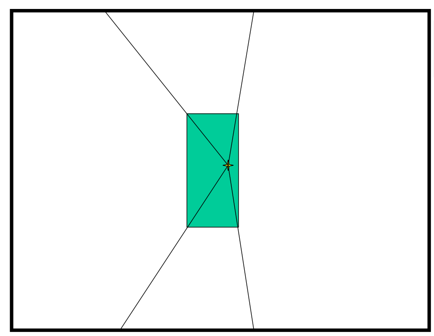
Left Camera

Another example of user input: vanishing point and back face of view volume are defined



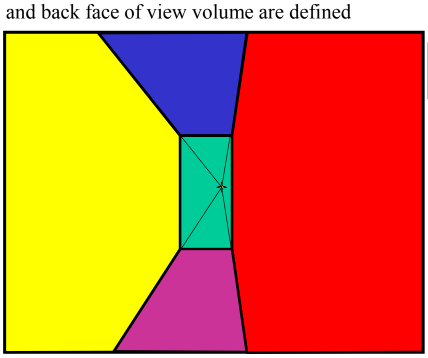
Left Camera

Another example of user input: vanishing point and back face of view volume are defined



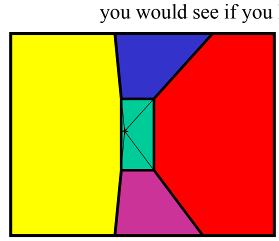
Right Camera

Another example of user input: vanishing point and back face of view volume are defined

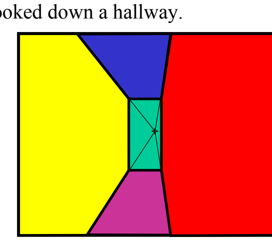


Right Camera

Comparison of two camera placements – left and right. Corresponding subdivisions match view you would see if you looked down a hallway.

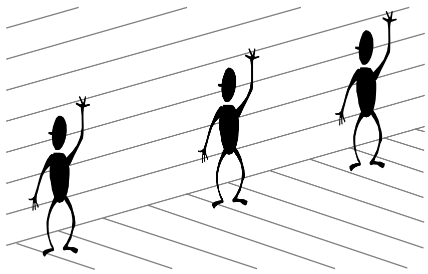


Left Camera

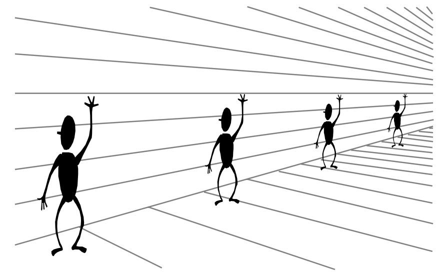


Right Camera

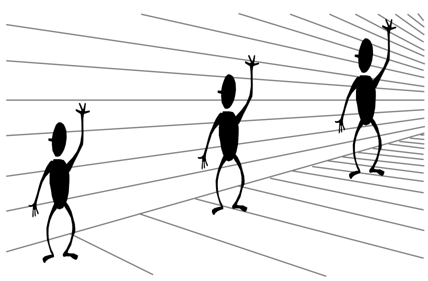
Comparing heights



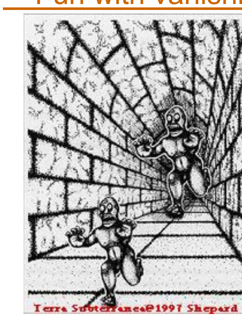
Perspective cues

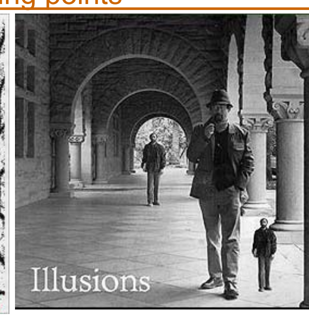


Perspective cues



Fun with vanishing points

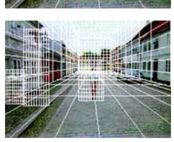




"Tour into the Picture" (SIGGRAPH '97)

Create a 3D "theatre stage" of five billboards

Specify foreground objects through bounding polygons



Use camera transformations to navigate through the scene

The idea

Many scenes (especially paintings), can be represented as an axis-aligned box volume (i.e. a stage)

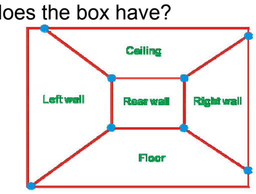
Key assumptions:

- All walls of volume are orthogonal
- Camera view plane is parallel to back of volume
- Camera up is normal to volume bottom

How many vanishing points does the box have?

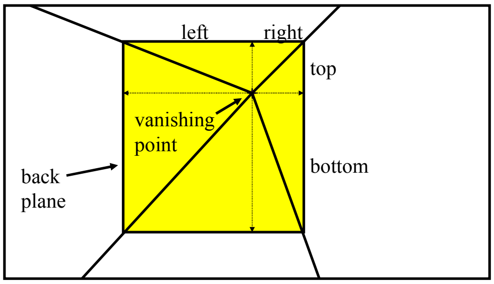
- Three, but two at infinity
- Single-point perspective

Can use the vanishing point to fit the box to the particular Scene!



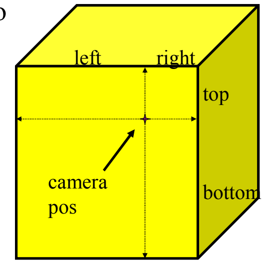
2D to 3D conversion

First, we can get ratios

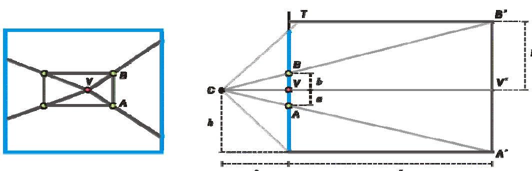


2D to 3D conversion

- Size of user-defined back plane must equal size of camera plane (orthogonal sides)
- Use top versus side ratio to determine relative height and width dimensions of box
- Left/right and top/bot ratios determine part of 3D camera placement



Depth of the box



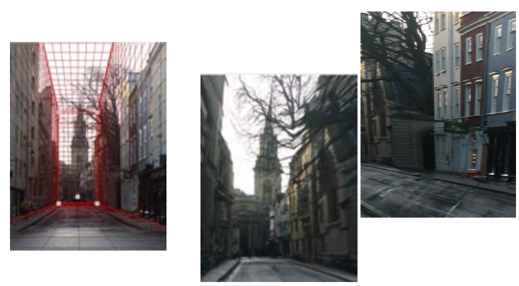
Can compute by similar triangles (CVA vs. CV'A')
Need to know focal length f (or FOV)

Note: can compute position on any object on the ground
- Simple unprojection
- What about things off the ground?

DEMO

Now, we know the 3D geometry of the box

We can texture-map the box walls with texture from the image



Foreground Objects

Use separate billboard for each

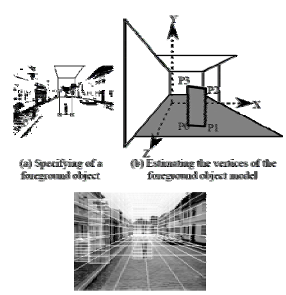
For this to work, three separate images used:
- Original image.
- Mask to isolate desired foreground images.
- Background with objects removed

Foreground Objects

Add vertical rectangles for each foreground object

Can compute 3D coordinates P0, P1 since they are on known plane.

P2, P3 can be computed as before (similar triangles)



(a) Specifying of a foreground object
(b) Estimating the vertices of the foreground object model
(c) Three foreground object models

Quiz: which is 1,2,3-point perspective

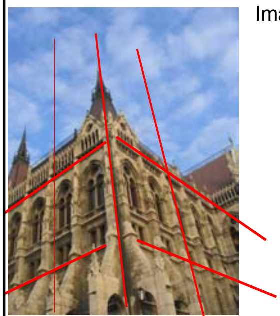
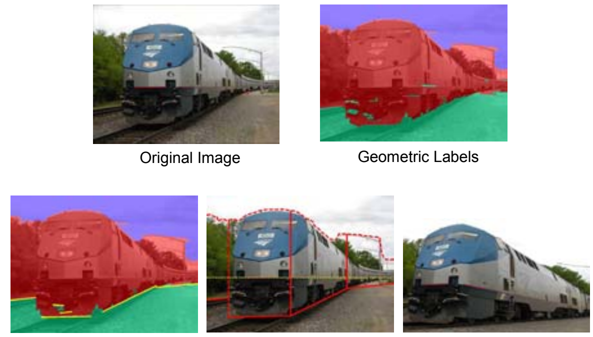


Image A

Image B

Image C

Automatic Photo Pop-up



Original Image

Geometric Labels

Fit Segments

Cut and Fold

Novel View

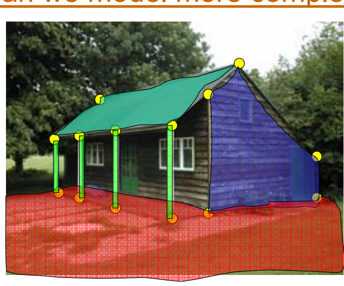
Foreground DEMO

Results

Input Image

Automatic Photo Pop-up

How can we model more complex scene?

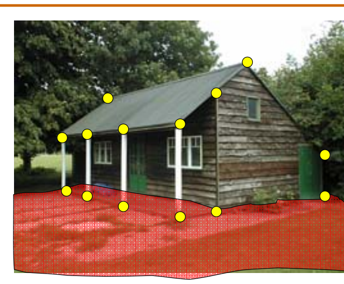


Find world coordinates (X,Y,Z) for a few points

Connect the points with planes to model geometry

- Texture map the planes

Finding world coordinates (X,Y,Z)

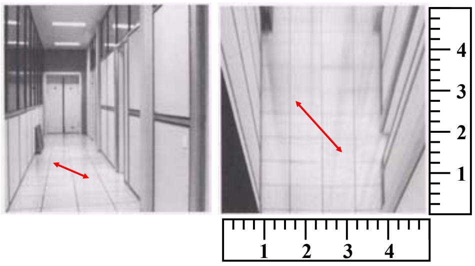


Define the ground plane (Z=0)

Compute points (X,Y,0) on that plane

Compute the *heights* Z of all other points

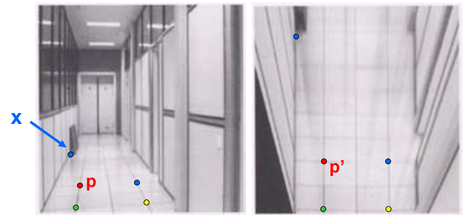
Measurements on planes



Approach: unwarp, then measure

What kind of warp is this?

Unwarp ground plane



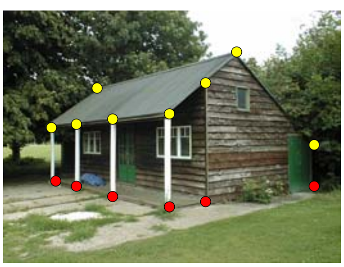
Our old friend – the homography

Need 4 reference points with world coordinates

p = (x,y)

p' = (X,Y,0)

Finding world coordinates (X,Y,Z)



Define the ground plane (Z=0)

Compute points (X,Y,0) on that plane

Compute the *heights* Z of all other points

Preliminaries: projective geometry

Ames Room

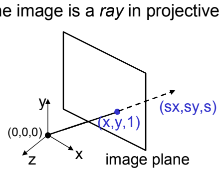
The projective plane

Why do we need homogeneous coordinates?

- represent points at infinity, homographies, perspective projection, multi-view relationships

What is the geometric intuition?

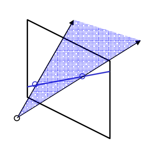
- a point in the image is a *ray* in projective space



- Each *point* (x,y) on the plane is represented by a *ray* (sx,sy,s)
  - all points on the ray are equivalent: (x, y, 1) ≅ (sx, sy, s)

Projective lines

What does a line in the image correspond to in projective space?



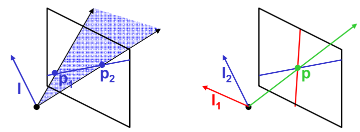
- A line is a *plane* of rays through origin
  - all rays (x,y,z) satisfying: ax + by + cz = 0

$$\text{in vector notation: } \quad 0 = \underset{\mathbf{l}}{\begin{bmatrix} a & b & c \end{bmatrix}} \underset{\mathbf{p}}{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}$$

- A line is also represented as a homogeneous 3-vector **l**

Point and line duality

- A line **l** is a homogeneous 3-vector
- It is $\perp$ to every point (ray) **p** on the line: **l p**=0



What is the line **l** spanned by rays $\mathbf{p_1}$ and $\mathbf{p_2}$?

- **l** is $\perp$ to $\mathbf{p_1}$ and $\mathbf{p_2}$ $\Rightarrow$ $\mathbf{l} = \mathbf{p_1} \times \mathbf{p_2}$
- **l** is the plane normal

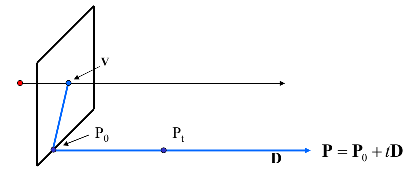
What is the intersection of two lines $\mathbf{l_1}$ and $\mathbf{l_2}$?

- **p** is $\perp$ to $\mathbf{l_1}$ and $\mathbf{l_2}$ $\Rightarrow$ $\mathbf{p} = \mathbf{l_1} \times \mathbf{l_2}$

Points and lines are *dual* in projective space

- given any formula, can switch the meanings of points and lines to get another formula

Computing vanishing points

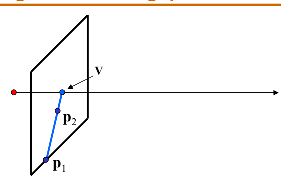


$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} P_X/t + D_X \\ P_Y/t + D_Y \\ P_Z/t + D_Z \\ 1/t \end{bmatrix} \qquad t \to \infty \qquad \mathbf{P}_\infty \cong \begin{bmatrix} D_X \\ D_Y \\ D_Z \\ 0 \end{bmatrix}$$

Properties $\quad \mathbf{v} = \mathbf{\Pi P}_\infty$

- $\mathbf{P}_\infty$ is a point at *infinity*, **v** is its projection
- They depend only on line *direction*
- Parallel lines $\mathbf{P}_0$ + t**D**, $\mathbf{P}_1$ + t**D** intersect at $\mathbf{P}_\infty$

Computing vanishing points



What is the line **l** spanned by rays $\mathbf{p_1}$ and $\mathbf{p_2}$?

- **l** is $\perp$ to $\mathbf{p_1}$ and $\mathbf{p_2}$ $\Rightarrow$ $\mathbf{l} = \mathbf{p_1} \times \mathbf{p_2}$
- **l** is the plane normal

What is the intersection of two lines $\mathbf{l_1}$ and $\mathbf{l_2}$?

- **v** is $\perp$ to $\mathbf{l_1}$ and $\mathbf{l_2}$ $\Rightarrow$ $\mathbf{v} = \mathbf{l_1} \times \mathbf{l_2}$

What is the intersection of a set of lines $\mathbf{l_1}$, $\mathbf{l_i}$ … $\mathbf{l_n}$?

$$\mathbf{M} = \sum_i \mathbf{l}_i \mathbf{l}_i^T$$

Eigenvector of M with smalest eigenvalues is v

Vanishing Points and Projection Matrix

Camera Projection Matrix

- $\mathbf{v} = \mathbf{\Pi X} = [\boldsymbol{\pi}_1 \quad \boldsymbol{\pi}_2 \quad \boldsymbol{\pi}_3 \quad \boldsymbol{\pi}_4]\mathbf{X}$
- $\boldsymbol{\pi}_1 = \mathbf{\Pi}[1 \quad 0 \quad 0 \quad 0]^T$ = X vanishing point ($\mathbf{v}_\mathrm{X}$)
- similarly, $\boldsymbol{\pi}_2 = \mathbf{v}_\mathrm{Y}$, $\boldsymbol{\pi}_3 = \mathbf{v}_\mathrm{Z}$
- $\boldsymbol{\pi}_4 = \mathbf{\Pi}[0 \quad 0 \quad 0 \quad 1]^T$ = projection of world origin

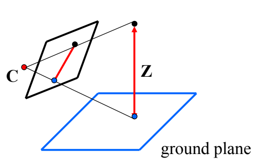
$\rightarrow$ convenient to choose $\boldsymbol{\pi}_4 = \dfrac{\mathbf{v}_X \times \mathbf{v}_Y}{\|\mathbf{v}_X \times \mathbf{v}_Y\|}$ call this **l**

$$\mathbf{\Pi} = [\mathbf{v}_X \quad \mathbf{v}_Y \quad \mathbf{v}_Z \quad \mathbf{l}]$$

Not So Fast! We only know **v**'s up to a scale factor

$$\mathbf{\Pi} = [a\,\mathbf{v}_X \quad b\mathbf{v}_Y \quad \alpha\mathbf{v}_Z \quad \mathbf{l}]$$

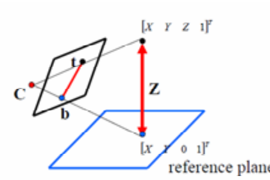
Measuring height without a ruler



Compute Z from image measurements

- Need more than vanishing points to do this

Measuring Heights



Compute Z from Image Measurements

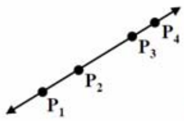
- Will actually calculate αZ (scaled height)
  - can convert to actual (Euclidean) height given a reference point
- First geometric argument
- Then algebraic derivation and formula

The Cross Ratio

A Projective Invariant

- Something that does not change under projective transformations (including perspective projection)

The Cross-Ratio of 4 Colinear Points



$$\frac{(P_3 - P_1)(P_4 - P_2)}{(P_3 - P_2)(P_4 - P_1)}$$

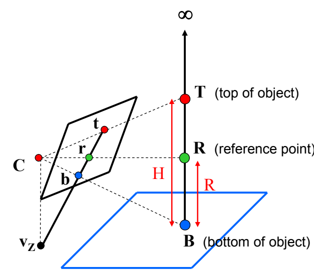
Can permute the point ordering

- 4! = 24 different invariants

This is the fundamental invariant of projective geometry

- likely that all other invariants derived from cross-ratio

Measuring height



$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|} = \frac{H}{R}$$

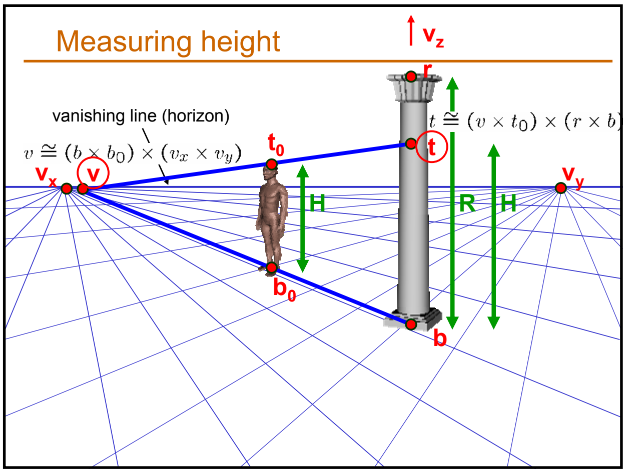
scene cross ratio

$$\frac{\|\mathbf{t}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|} = \frac{H}{R}$$

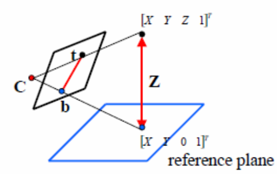
image cross ratio

scene points represented as $\mathbf{P} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$ image points as $\mathbf{p} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Measuring height



Measuring Height



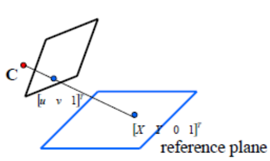
Algebraic Derivation

- $\rho\,\mathbf{b} = \mathbf{\Pi}[X \quad Y \quad 0 \quad 1]^T = Xa\,\mathbf{v}_X + Yb\,\mathbf{v}_Y + \mathbf{l}$
- $\mu\,\mathbf{t} = \mathbf{\Pi}[X \quad Y \quad Z \quad 1]^T = Xa\,\mathbf{v}_X + Yb\,\mathbf{v}_Y + \alpha Z\,\mathbf{v}_Z + \mathbf{l}$
- Eliminating ρ and μ yields

$$\alpha Z = \frac{-\|\mathbf{b}\times\mathbf{t}\|}{\mathbf{l}^T\mathbf{b}\,\|\mathbf{v}_Z \times \mathbf{t}\|}$$

- Can calculate α given a known height in scene

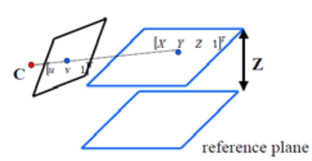
Measurements Within Reference Plane



Planar Perspective Map (*homography*) **H**

- **H** Maps reference plane X-Y coords to image plane u-v coords
- Fully determined from 4 known points on ground plane
  - Option A: physically measure 4 points on ground
  - Option B: find a square, guess the size
  - Option C: Note $\mathbf{H} = [a\mathbf{v}_X \;\; b\mathbf{v}_Y \;\; \mathbf{l}]$ (columns 1,2,4 of Π)
    - » play with scale factors a and b until the model "looks right"
- Given u-v, can find X-Y by $\mathbf{H}^{-1}$

Measurements Within Parallel Plane



Planar Perspective Map (*homography*) $\mathbf{H}_Z$

- $\mathbf{H}_Z$ Maps X-Y-Z coords to image plane u-v coords

$$\mathbf{H}_Z = [a\mathbf{v}_X \quad b\mathbf{v}_Y \quad \alpha Z\,\mathbf{v}_Z + \mathbf{l}]$$

- Given u-v, can find X-Y by $\mathbf{H}_Z^{-1}$
- Another way is to first map parallel plane to reference plane:
  - parallel planes related by a *homology* (5 parameter homography)
  - $\hat{\mathbf{H}} = \mathbf{H}_Z\,\mathbf{H}^{-1} = \mathbf{I} + \alpha Z\,\mathbf{v}_Z{}^T\mathbf{l}$
  - maps u-v coords on parallel plane to u-v coords on ref. plane

Assignment 4 Due: Never

Implement Technique in Criminisi et al.

- Load in an image
- Click on parallel lines defining X, Y, and Z directions
- Compute vanishing points
- • Specify points on reference plane, ref. height
- Compute 3D positions of several points
- Create a 3D model from these points
- Extract texture maps
  - using Assignment 2 warping code
- Output a VRML model