Single View Metrology

A. Criminisi, I. Reid and A. Zisserman (ICCV 99)

Make scene measurements from a single image

- Application: 3D from a single image

Assumptions

1. 3 orthogonal sets of parallel lines
2. 4 known points on ground plane
3. 1 height in the scene

Can still get an *affine reconstruction* without 2 and 3

Preliminaries--Projective Geometry

Vanishing Points

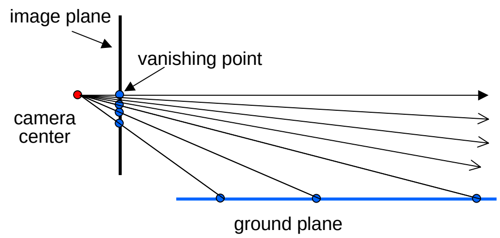
Points at Infinity

Vanishing Lines

The Cross-Ratio

All quantities on these slides are in homogeneous coordinates except when specified otherwise

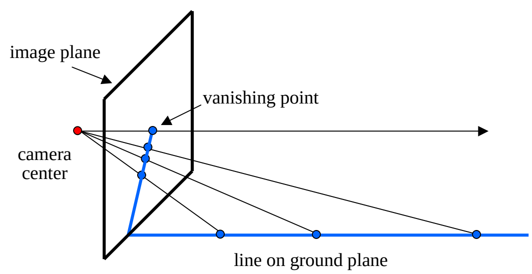
Vanishing Points (1D)



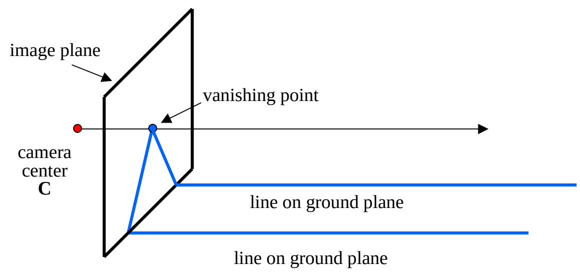
Vanishing point

- projection of a point at infinity

Vanishing Points (2D)



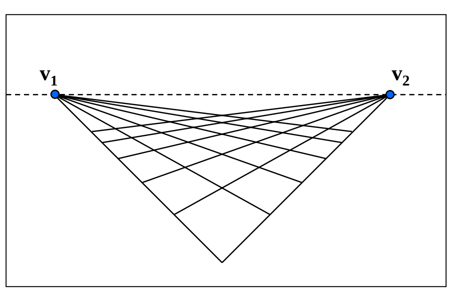
Vanishing Points



Properties

- Any two parallel lines on same plane have the same vanishing point
- Any two parallel lines PERIOD have the same vanishing point
- The ray from **C** through the vanishing point is parallel to the line
- An image may have more than one vanishing point

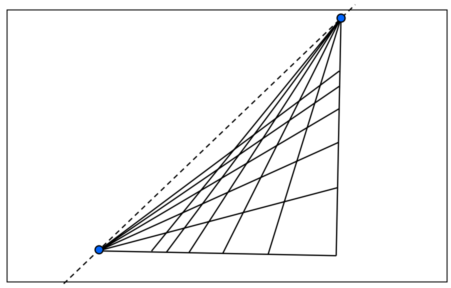
Vanishing Lines



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*
  - called *vanishing line* in the Criminisi paper
- Note that different planes define different vanishing lines

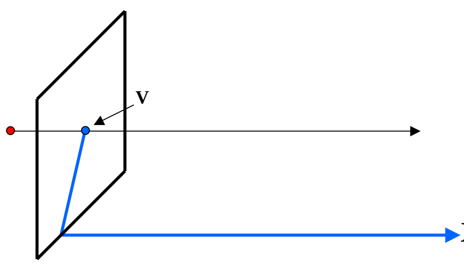
Vanishing Lines



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*
  - called *vanishing line* in the Criminisi paper
- Note that different planes define different vanishing lines

Computing Vanishing Points

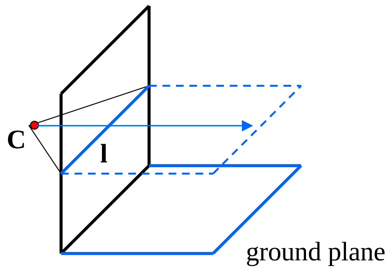


$$\mathbf{X}(\mathrm{t}) = \begin{bmatrix} A_X + tD_X \\ A_Y + tD_Y \\ A_Z + tD_Z \\ 1 \end{bmatrix} = \begin{bmatrix} A_X / t + D_X \\ A_Y / t + D_Y \\ A_Z / t + D_Z \\ 1/t \end{bmatrix} \qquad t \to \infty \qquad \mathbf{X}_\infty = \begin{bmatrix} D_X \\ D_Y \\ D_Z \\ 0 \end{bmatrix}$$

Properties $\mathbf{v} = \mathbf{\Pi X}_\infty$

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

Computing Vanishing Lines

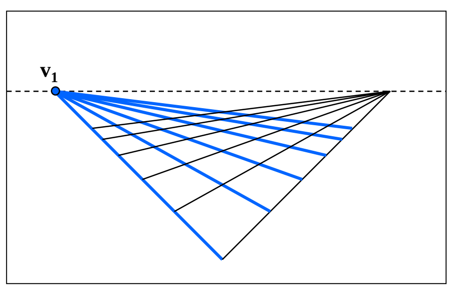


Properties

- **l** is intersection of horizontal plane through **C** with image plane
- Compute **l** from two sets of parallel lines on ground plane
- All points at same height as **C** project to **l**
- Provides way of comparing height of objects in the scene

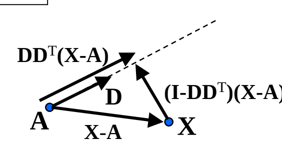
Computing Vanishing Points from Parallelism

Better method by Bob Collins: see course web page



Coordinates in this slide are *NOT* homogenous

Intersection of Two or More Lines



- Lines $\mathbf{A}_1 + t\mathbf{D}_1, \ldots, \mathbf{A}_n + t\mathbf{D}_n$ ($\mathbf{D}_i$ unit vectors)
- Minimize sum squared distance to all lines

$$\text{minimize} \sum_{i=1}^{n} (\mathbf{X}-\mathbf{A}_i)^T \underbrace{(\mathbf{I}-\mathbf{D}_i\mathbf{D}_i^T)^T(\mathbf{I}-\mathbf{D}_i\mathbf{D}_i^T)}_{\mathbf{C}_i}(\mathbf{X}-\mathbf{A}_i)$$

$$\text{differentiating wrt. } \mathbf{X} \text{ and setting to } 0: \quad \left(\sum_{i=1}^{n} \mathbf{C}_i\right)\mathbf{X} = \sum_{i=1}^{n} \mathbf{C}_i \mathbf{A}_i$$

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 ($\mathbf{v}_X$)
- similarly, $\boldsymbol{\pi}_2 = \mathbf{v}_Y$, $\boldsymbol{\pi}_3 = \mathbf{v}_Z$
- $\boldsymbol{\pi}_4 = \mathbf{\Pi}[0 \quad 0 \quad 0 \quad 1]^T$ = 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 $\mathbf{l}$

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

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

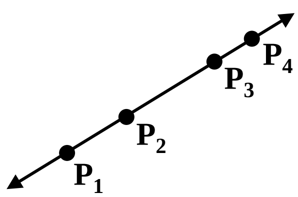
$$\mathbf{\Pi} = [a\,\mathbf{v}_X \quad b\mathbf{v}_Y \quad \alpha\mathbf{v}_Z \quad \mathbf{l}]$$

The Cross Ratio

A Projective Invariant

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

The Cross-Ratio of 4 Colinear Points



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

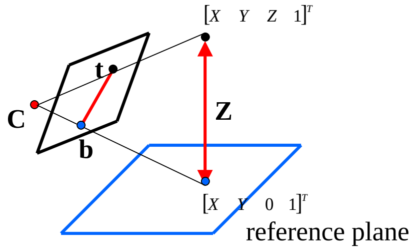
Can permute the point ordering

- 4! = 24 different invariants

This is the fundamental invariant of projective geometry

- likely that all other invariants derived from cross-ratio

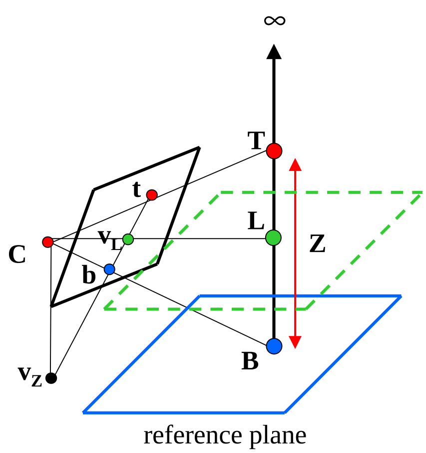
Measuring Height



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

Measuring Height



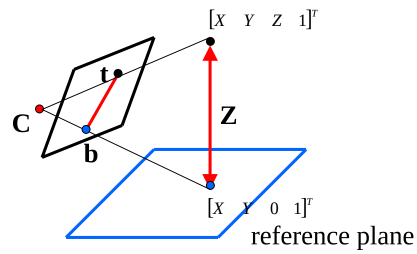
Scene Cross Ratio

$$\frac{(T-B)(\infty-L)}{(T-L)(\infty-B)} = \frac{1}{T-L}Z = \alpha Z$$

Image Cross Ratio

$$\frac{(t-b)(v_Z-v_L)}{(t-v_L)(v_Z-b)} = \alpha Z$$

Measuring Height



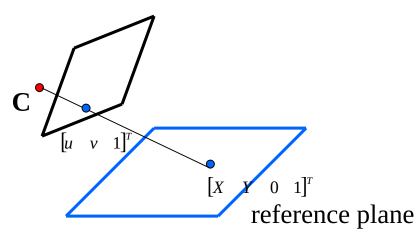
Algebraic Derivation

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

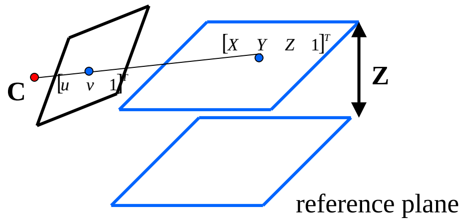
Measurements Within Reference Plane



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 **H** = [a$\mathbf{v}_X$ b$\mathbf{v}_Y$ **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)
  - $\hat{\mathbf{H}} = \mathbf{H}_Z\, \mathbf{H}^{-1} = I + \alpha Z\, \mathbf{v}_Z^T\, \mathbf{l}$
  - maps u-v coords on parallel plane to u-v coords on ref. plane

Assignment 3: Single View Modeling

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 1 warping code
- Output a VRML model