4. Single View Geometry

Computer Vision

Zoltan Kato

http://www.inf.u-szeged.hu/~kato/

Slides adopted from

CS 395/495-26: Spring 2004

IBMR: Intro to P^3

3-D Projective Geometry

Jack Tumblin

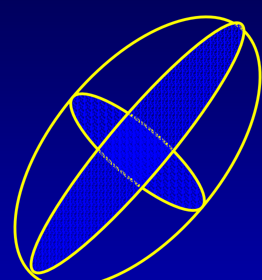
jet@cs.northwestern.edu

Quadrics Summary

Quadrics are the 'x² family' in **P^3:**

- Point Quadric: $\mathbf{x}^T Q\, \mathbf{x} = 0$
- Plane Quadric: $\pi^T Q^* \pi = 0$

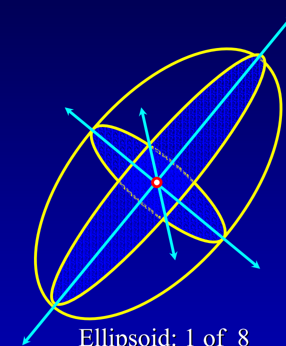
- Transformed Quadrics:
  - Point Quadric: $Q' = H^{-T} Q\, H$
  - Plane Quadric: $Q^{*\prime} = H\, Q^*\, H^T$
- Symmetric Q, Q* matrices:
  - **10** parameters but **9 DOF**; 9 points or planes
  - (or less if degenerate…)
  - 4x4 symmetric, so SVD(**Q**) = $\mathbf{USU}^T$



Ellipsoid: 1 of 8 quadric types

Quadrics Summary

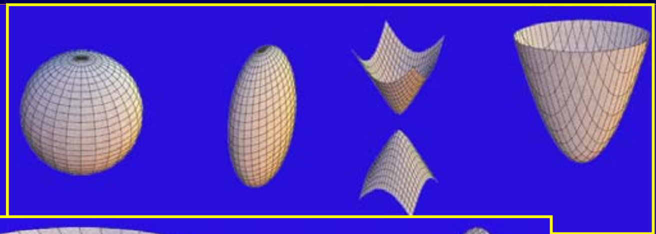
- SVD(Q) =$\mathbf{USU}^T$:
  - **U** columns are quadric's **axes**
  - **S** diagonal elements: scale
- On U axes, write any quadric as:
  $au_1^2 + bu_2^2 + cu_3^2 + d = 0$
- Classify quadrics by
  - sign of a,b,c,d: (>0, 0, <0)
- Book's method:
  - scale a,b,c,d to (+1, 0, –1)
  - classify by Q's **rank** and (**a+b+c+d**)



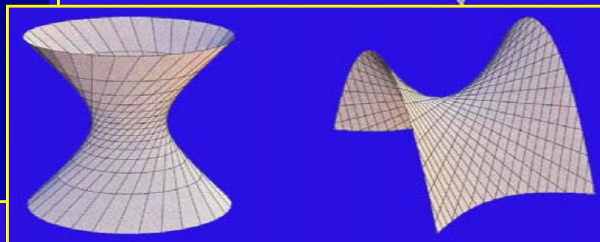
Ellipsoid: 1 of 8 quadric types

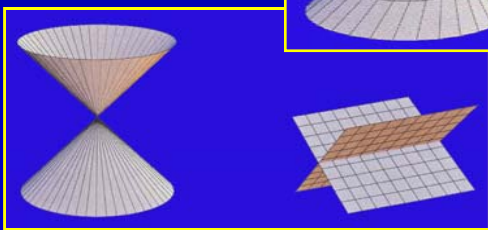
Quadrics Summary

All Unruled Quadrics are Rank 4: (See page 55)



BUT Some Rank 4 quadrics are Ruled:





and All degenerate quadrics (Rank<4) are Ruled (or Conic)

New Weirdness: Absolute Conic Ω_∞

- **WHY learn Ω_∞?** Similar to C_∞ for P^2…
  - Angles from directions (d_1, d_2) or planes (π_1, π_2)
  - π_∞ has 3DOF for H_P; Ω_∞ has 5DOF for H_A
- Ω_∞ Requires TWO equations:

Ω_∞ : $x_1^2 + x_2^2 + x_3^2 = 0,$	or '2D point conic where C = I'
$x_4 = 0,$	or 'all points are on π_∞'

- Ω_∞ is complex 2D Point Conic on the π_∞ plane

  Recall plane at infinity $\pi_\infty = [0, 0, 0, 1]^T$

  holds 'directions' $d = [x_1, x_2, x_3, 0]^T$

New Weirdness: Absolute Conic Ω_∞

- Ω_∞ is complex 2D Point Conic on the π_∞ plane

$x_1^2 + x_2^2 + x_3^2 = 0,$	or '2D point conic where C = I'
$x_4 = 0,$	or 'all points are on π_∞'

- Only $H_A H_P$ transforms Ω_∞ (stays Ω_∞ for H_S)
- All circles (in any π) intersect Ω_∞ circular pts.
  - (recall: circular pts. hold 2 axes: $x \pm iy$)
- All spheres (in P^3) intersect π_∞ at all Ω_∞ pts.

New Weirdness: Absolute Conic Ω_∞

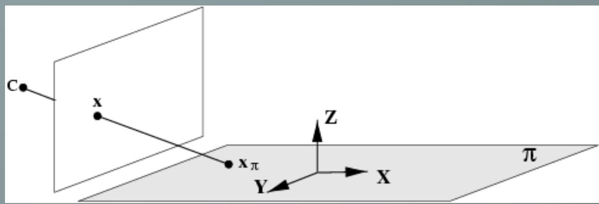
Ω_∞ measures angles between Directions (d_1, d_2)

- World-space Ω_∞ is I_{3x3} (ident. matrix) within π_∞
- Image-space Ω_∞' is transformed
- Euclidean world-space angle θ is given by:

$$\cos(\theta) = \frac{(d_1^T \Omega_\infty' d_2)}{\sqrt{(d_1^T \Omega_\infty' d_1)(d_2^T \Omega_\infty' d_2)}}$$

- Directions d_1, d_2 are orthogonal iff $d_1^T \Omega_\infty' d_2 = 0$

Action of projective camera on planes



$$x = PX = [p_1 p_2 p_3 p_4]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = [p_1 p_2 p_4]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

- The most general transformation that can occur between a scene plane and an image plane under perspective imaging is a plane projective transformation
  - Assume world coordinate system is aligned with the plan π (➔ Z=0)
- affine camera ➔ affine transformation

Action of projective camera on lines

- forward projection

$$X(\mu) = P(A + \mu B) = PA + \mu PB = a + \mu b$$

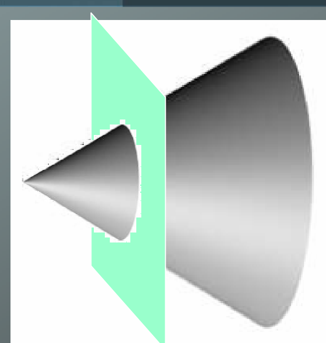
- back-projection

$$\Pi = P^T l$$
$$\Pi^T X = l^T P X$$

Action of projective camera on conics

- back-projection to cone

$$Q_{co} = P^T C P \quad x^T C x = X^T P^T C P X = 0$$

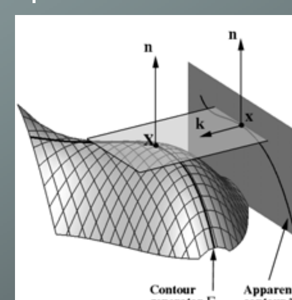


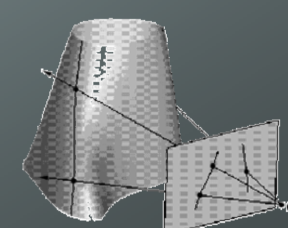
- example:

$$Q_{co} = \begin{bmatrix} K^T \\ 0 \end{bmatrix} C^T [K \mid 0] = \begin{bmatrix} K^T C K & 0 \\ 0 & 0 \end{bmatrix}$$

Images of smooth surfaces

- The contour generator Γ is the set of points **X** on **S** at which rays are tangent to the surface.
  - The corresponding apparent contour γ is the set of points **x** which are the image of **X**, i.e. γ is the image of Γ
  - The contour generator Γ depends only on position of projection center
  - γ depends also on rest of **P**

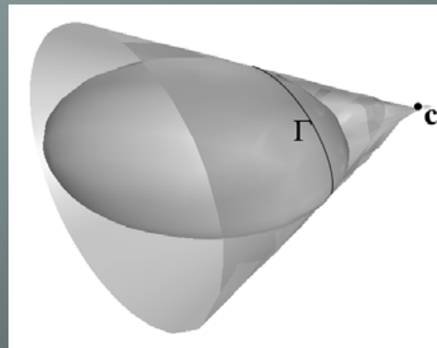




Action of projective camera on quadrics

- back-projection to cone

$$C^* = PQ^*P^T$$
$$\Pi^T Q^* \Pi = l^T P Q^* P^T l = 0$$



- The plane of Γ for a quadric **Q** and camera center **C** is given by **P=QC** (follows from pole-polar relation)

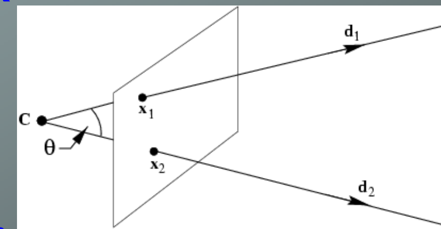
$$Q_{CO} = (V^T Q V)Q - (QV)(QV)^T$$

- The cone with vertex **V** and tangent to the quadric **Q** is

$$Q_{CO} V = 0$$

What does calibration give?

- The camera calibration matrix **K** is the (affine) transformation between **x** and the ray's direction **d=K^{-1}x** measured in the camera's Euclidean coordinate frame
  - In general, **d** is ***not*** a unit vector
  - The angle between two rays **d_1**, **d_2** corresponding to image points **x_1**, **x_2** may be obtained by the cosine formula (angle between two vectors).



$$x = K[I \mid 0]\begin{bmatrix} d \\ 0 \end{bmatrix}$$

- Calibrated camera = direction sensor

$$d = K^{-1}x$$

$$\cos\theta = \frac{d_1^T d_2}{\sqrt{(d_1^T d_1)(d_2^T d_2)}} = \frac{x_1^T (K^{-T}K^{-1}) x_2}{\sqrt{(x_1^T (K^{-T}K^{-1}) x_1)(x_2^T (K^{-T}K^{-1}) x_2)}}$$

What does calibration give?

- An image line **l** defines a plane through the camera center with normal **n=K^{T}l** measured in the camera's Euclidean frame
  - In general, **n** is not a unit vector

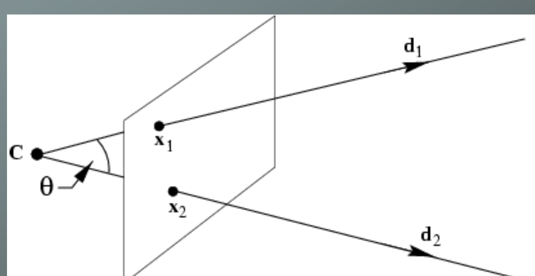


Image of the absolute conic

- mapping between **p∞** to an image is given by the planar homogaphy **x=Hd**, with **H=KR**

$$x = PX_\infty = KR[I \mid -\tilde{C}]\begin{pmatrix} d \\ 0 \end{pmatrix} = KRd$$

- Since the absolute conic Ω_∞ is on **p∞**, its image (IAC) under **H** is given by

$$\omega = (KK^T)^{-1} = K^{-T}K^{-1}$$

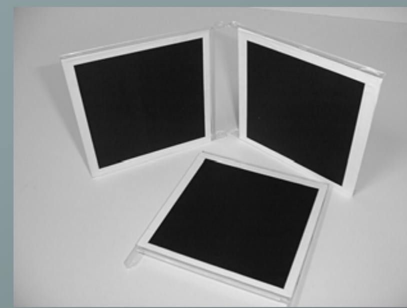
- Note that ω is an imaginary point conic with no real points.
  - It is a convenient algebraic device…

Image of the absolute conic

$$\omega = (KK^T)^{-1} = K^{-T}K^{-1}$$

- IAC depends only on intrinsics **K**
- angle between two rays ⟹ $\cos\theta = \frac{x_1^T \omega x_2}{\sqrt{(x_1^T \omega x_1)(x_2^T \omega x_2)}}$
  - If x_1 and x_2 correspond to orthogonal directions, then $x_1 \omega x_2 = 0$
- D(ual)IAC= $\omega^* = KK^T$
- Once ω (or ω^*) is identified ➔ **K** may be obtained by Cholesky factorisation
- image of circular points:
  - A plane **p** intersects **p∞** in a line
  - This line intersects Ω_∞ in two points (circular points of **p**)
  - The image of these points lie on ω where the vanishing line of **p** intersects ω

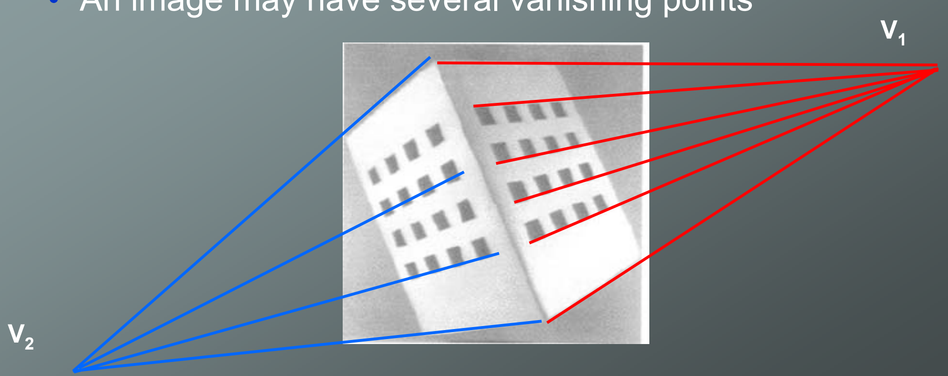
A simple calibration device (Z. Zhang)



- Image of 3 squares on planes
  - Which are not parallel
  - Not necessarily orthogonal
- Provides sufficient constraints to compute **K**
- Compute H for each square
  - corners (0,0), (1,0), (0,1), (1,1)
  - The alignment of the plane coordinate system with the square is a ***similarity transform*** ➔ does not affect circular point's position on the plane
- Compute the imaged circular points **$H(1,\pm i,0)^T$**
  - **$h_1 \pm ih_2$**
- Fit a conic to 6 circular points:
  - $(h_1 \pm ih_2)^T \omega (h_1 \pm ih_2) = 0$
- Compute **K** from ω through Cholesky factorization

Vanishing points

- All parallel lines in 3D space appear to meet in a point on the image - the **vanishing point**
  - common intersection of the image lines
  - An image may have several vanishing points



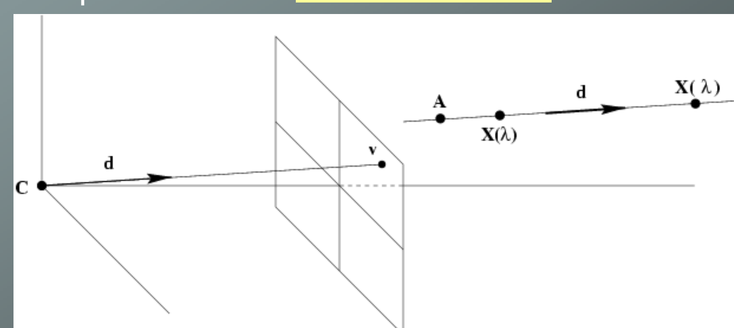
Vanishing points

- All 3D lines with the same direction intersect at **p∞** in the same point.
- The vanishing point is simply the image of this point.

$$x(\lambda) = PX(\lambda) = PA + \lambda PD = a + \lambda Kd$$

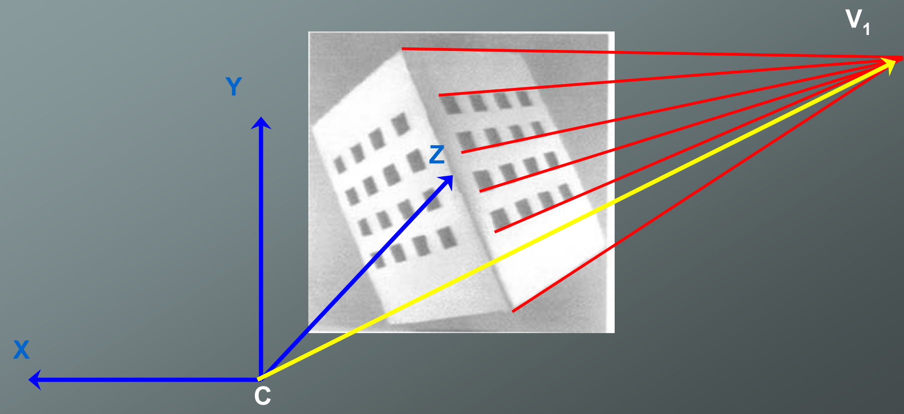
$$v = \lim_{\lambda\to\infty} x(\lambda) = \lim_{\lambda\to\infty} (a + \lambda Kd) = Kd$$

$$v = PX_\infty = Kd$$



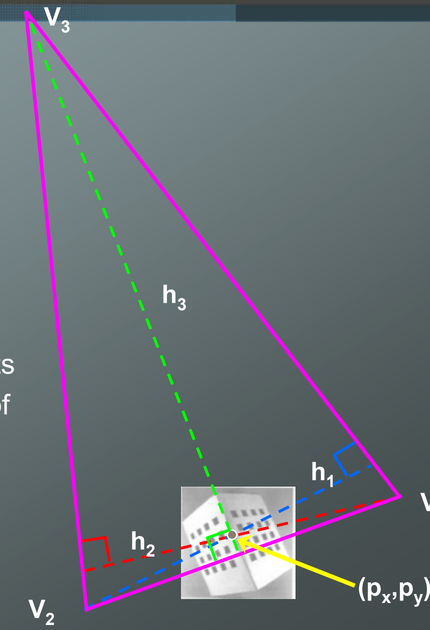
Important property

- Vector CV$_1$ (from the center of projection to the vanishing point) is parallel to the parallel lines

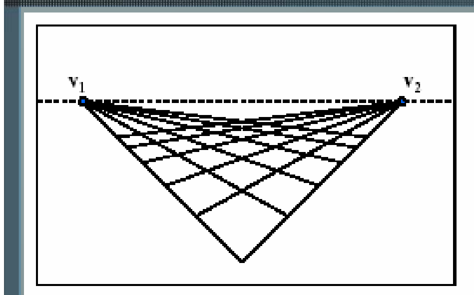


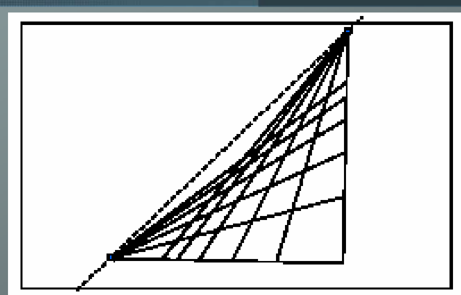
Orthocenter theorem

- Three orthogonal sets of parallel lines can be used to determine the image center without any information about focal length and extrinsic parameters:
  - **Input**: three mutually orthogonal sets of parallel lines in an image
  - **T**: a triangle on the image plane defined by the three vanishing points
  - Image center (p_x,p_y) = orthocenter of triangle **T**
    - Orthocenter of a triangle is the common intersection of the three altitudes



Vanishing lines

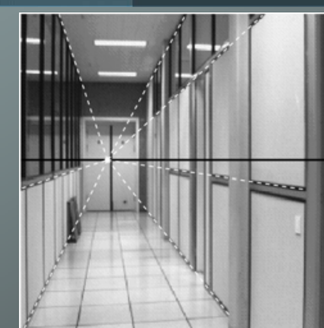




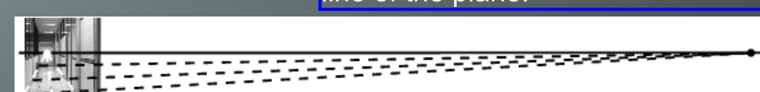
- Any set of parallel lines on the plane define a vanishing point
- The union of all of these vanishing points is the ***vanishing*** (or horizon) ***line***
- Note that the vanishing line depends only on the ***orientation***
  - Differently oriented planes define different vanishing lines
  - Parallel planes share the same vanishing line (which is the image of the parallel 3D plane's intersection at **p∞**)

Vanishing lines

The vanishing line of the ground plane (the horizon) may be obtained from two sets of parallel lines on the plane



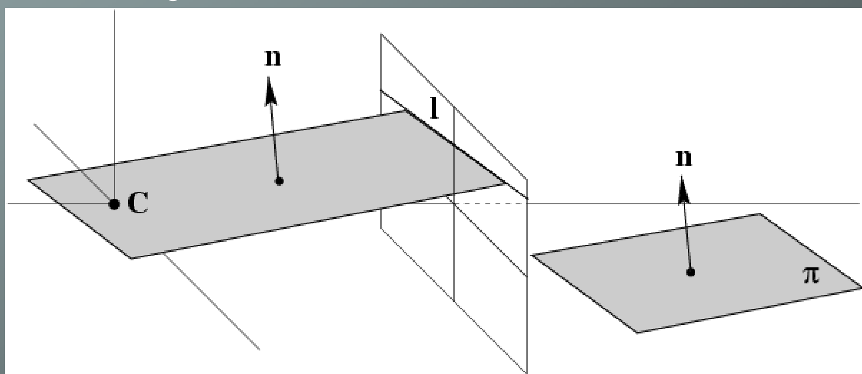
The vanishing point of lines parallel to the ground plane lies on the vanishing line of the plane.



The vanishing points of lines nearly parallel to the image plane are distant from the actual image.

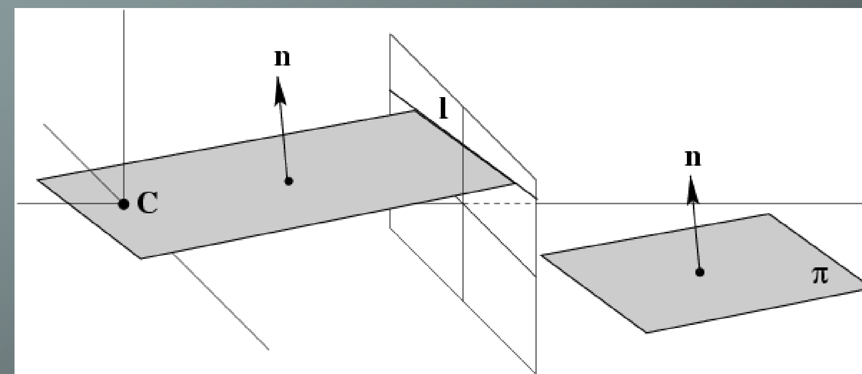
Vanishing lines

- Geometry:
  - The vanishing line **l** is constructed by intersecting the image with a plane
    - Parallel to the scene plane π
    - Through the camera center **C**



Vanishing lines

- A plan through the camera center **C** with normal direction **n**
  - Intersects the image plane in the line $\mathbf{l}=\mathbf{K}^{-T}\mathbf{n}$
  - **l** is the vanishing line of planes perpendicular to **n**
- ➔ a plane with vanishing line **l** has orientation $\mathbf{n}=\mathbf{K}^{T}\mathbf{l}$ in the camera's Euclidean coordinate frame.



Vanishing lines

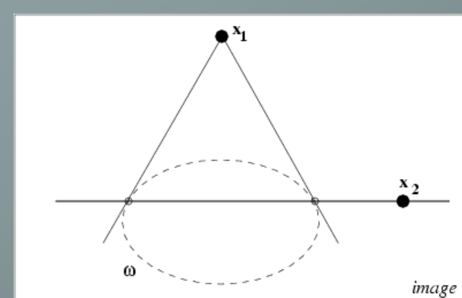
- The angle between two scene planes can be determined from their vanishing lines $\mathbf{l}_1$ and $\mathbf{l}_2$:

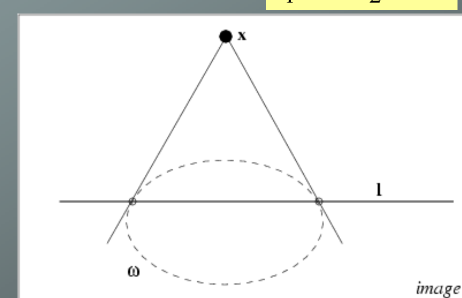
$$\cos\theta = \frac{v_1^T \omega v_2}{\sqrt{(v_1^T \omega v_1)(v_2^T \omega v_2)}}$$

- A scene plane may be metrically rectified given only its vanishing line:
  - The plane normal is known from **l**
  - ➔The camera can be rotated (synthetically) such that the plane becomes frontoparallel.
    - This is achieved by computing a homography.

Orthogonality and ω

- The vanishing lines of two perpendicular planes satisfy: $l_1^T \omega^* l_2 = 0$





- If x_1 and x_2 correspond to orthogonal directions, then $x_1 \omega x_2 = 0$
- They are ***conjugate*** with respect to ω

- A point **x** and a line **l** backprojecting to a line and a plane that are orthogonal are related by $\mathbf{l}=\omega\mathbf{x}$
- **x** and **l** are ***pole-polar*** with respect to ω

Calibration from vanishing points and lines

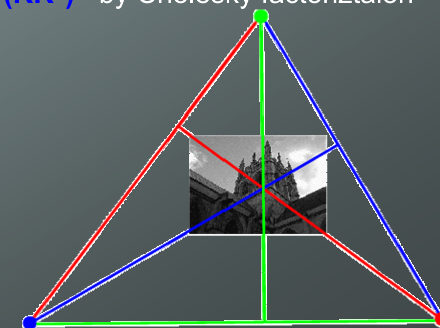
- Once ω is known, we can measure angle between rays
- ➔ If the angle between rays are known then a constraint is placed on ω
  - This is a quadratic constraint for arbitrary angles
  - Orthogonality results in a ***linear*** constraint
- Internal constraints may also be imposed
  - Zero skew: $s=K_{12}=0$ ➔ $\omega_{12}=\omega_{21}=0$
  - Square pixels: $s_x=K_{12}=K_{21}=s_y$ ➔ $\omega_{11}=\omega_{22}$

Calibration from vanishing points and lines

- Given three orthogonal vanishing point directions
  - Each pair gives $v_i^T \omega v_j = 0$
- + assume zero skew & square pixles
  - We have 5 constraints sufficient to compute ω

$$\omega = \begin{bmatrix} \omega_1 & 0 & \omega_2 \\ 0 & \omega_1 & \omega_3 \\ \omega_2 & \omega_3 & \omega_4 \end{bmatrix}$$

- Write the constraints by stacking them together to form the equation
  - **Aw=0**, where $w=[\omega_i]$ (i=1..4) and **A** is 3X4
  - Solve for **w** and compute **K** from $\omega=(KK^T)^{-1}$ by Cholesky factoriztaion followed by inversion.



Calibration from vanishing points and lines

- The principal point **p** can be obtained from the orthocenter
- The focal length
  - Consider the plane defined by **C**, **p** and one of the vanishing points (v_3)
  - The rays $C \rightarrow v_3$ and $C \rightarrow x$ are perpendicular to each other
  - The focal length is the distance of the image plane from **C**
  - by similar triangles: $f^2=d(p,v_3)d(p,x)$
- **Caution**: this method is degenerate if one of the vanishing points is at infinity (in that case **A** drops rank to 2!)