CAMERA CALIBRATION

SINGLE VIEW GEOMETRY
Motivation

• Good calibration is important when needed
  ▪ Reconstruct a world model
  ▪ Interact with the world
    • Robot, hand-eye coordination

We see a square of known size

Image plane

Evaluation of position of a square for 2 focal lengths (red and blue projection geometry)
Direct Linear Transform (DLT) 3D to 2D

Homogeneous coordinates
\( x = [x \ y \ w]^T \)
\( X = [X \ Y \ Z \ 1]^T \)

\[ \mathbf{x}_i = \mathbf{P} \mathbf{X}_i \]
\[ [\mathbf{x}_i] \times \mathbf{P} \mathbf{X}_i = 0 \]

\[ \mathbf{P} = \begin{bmatrix} \mathbf{P}_1^T \\ \mathbf{P}_2^T \\ \mathbf{P}_3^T \end{bmatrix} \]

\( \mathbf{P} \) is a 3x4 matrix in a 12 vector

Divide \( w_i \) ==> cartesian coor. 2D

\[ \begin{bmatrix} 0^T \\ w_i \mathbf{X}_i^T \\ -y_i \mathbf{X}_i^T \\ x_i \mathbf{X}_i^T \end{bmatrix} \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} = 0 \]

\[ A_i \mathbf{p} = 0 \]

\[ A \mathbf{p} = 0 \]

\( A \) has rank 11, \( \mathbf{p} \) right null-space
The algebraic error solution.

\[ \mathbf{A} \mathbf{p} = 0 \]

Minimal solution

\[ \mathbf{P} \text{ has 11 DOF, } 2 \text{ independent eq./points} \]

\[ \Rightarrow 5\frac{1}{2} \text{ correspondences needed (has to be 6)} \]

Over-determined solution

\[ n \geq 6 \text{ points} \]

minimize \[ \| \mathbf{A} \mathbf{p} \| \text{ subject to constraint} \]

\[ \| \mathbf{p} \| = 1 \text{ or the vector } \| \hat{\mathbf{p}}^3 \| = 1, \text{ where } \hat{\mathbf{p}}^3 \text{ is the vector } (p_{31}, p_{32}, p_{33})^T \]

\[ \Rightarrow \text{ use SVD } \mathbf{p} = \mathbf{V}_{-9} \]
Degenerate configurations of camera calibration

Points lie on plane and single line passing through projection center

Camera and points on a twisted cubic

Any twisted cubic is projectively equivalent.
Data normalization in DLT

Scale data to values of order 1.

1. move center of mass to origin
2. scale to yield order 1 values

\[
\tilde{x} = T x
\]

\[
\tilde{X} = U X
\]

\[
T = \begin{bmatrix}
\sigma_{2D} & 0 & \bar{x} \\
0 & \sigma_{2D} & \bar{y} \\
0 & 0 & 1
\end{bmatrix}^{-1}
\]

\[
U = \begin{bmatrix}
\sigma_{3D} & 0 & 0 & \bar{X} \\
0 & \sigma_{3D} & 0 & \bar{Y} \\
0 & 0 & \sigma_{3D} & \bar{Z} \\
0 & 0 & 0 & 1
\end{bmatrix}^{-1}
\]

(see also 2D estimation...)

\[\sqrt{2} \sqrt{3}\]
Improving $P$ Solution with Nonlinear Minimization

- Find $p$ using DLT -- algebraic distance
- Use as initialization for nonlinear minimization of $\sum_i d(x_i, PX_i^2)$ -- geometric distance
  - Use Levenberg-Marquardt iterative minimization
    - Canny edge detection
    - Straight line fitting to the detected edges
    - Intersecting the lines to obtain the images corners

Typically precision <1/10 of a pixel, if care is taken
Rule of thumb: 5 times the number of unknowns (11)
Objective
Given \( n \geq 6 \) world to image point correspondences \( \{X_i \leftrightarrow x_i\} \), determine the Maximum Likelihood estimate of the camera projection matrix \( P \), i.e. the \( P \) which minimizes \( \sum_i d(x_i, PX_i)^2 \).

Algorithm

(i) **Linear solution.** Compute an initial estimate of \( P \) using a linear method such as algorithm 4.2(p109):

(a) **Normalization:** Use a similarity transformation \( T \) to normalize the image points, and a second similarity transformation \( U \) to normalize the space points. Suppose the normalized image points are \( \tilde{x}_i = T x_i \), and the normalized space points are \( \tilde{X}_i = U X_i \).

(b) **DLT:** Form the \( 2n \times 12 \) matrix \( A \) by stacking the equations (7.2) generated by each correspondence \( \tilde{X}_i \leftrightarrow \tilde{x}_i \). Write \( p \) for the vector containing the entries of the matrix \( \tilde{P} \). A solution of \( A p = 0 \), subject to \( \|p\| = 1 \), is obtained from the unit singular vector of \( A \) corresponding to the smallest singular value.

(ii) **Minimize geometric error.** Using the linear estimate as a starting point minimize the geometric error (7.4):

\[
\sum_i d(\tilde{x}_i, \tilde{P} \tilde{X}_i)^2
\]

over \( \tilde{P} \), using an iterative algorithm such as Levenberg–Marquardt.

(iii) **Denormalization.** The camera matrix for the original (unnormalized) coordinates is obtained from \( \tilde{P} \) as

\[
P = T^{-1} \tilde{P} U.
\]

Algorithm 7.1. *The Gold Standard algorithm for estimating \( P \) from world to image point correspondences in the case that the world points are very accurately known.*
Roger Y. Tsai
IEEE Robotics and Automation
August 1987

Takes care of radial distortions too... see later.
Calibration Procedure

• Calibration target: 2 planes at right angle with checkerboard patterns (Tsai grid)
  § We know positions of pattern corners only with respect to a coordinate system of the world.
  § Position camera in front of target and find images of corners.
  § Obtain equations that describe imaging and contain internal parameters of camera.
    • As a side benefit, we find position and orientation of camera with respect to target (camera \textit{pose})
      exterior orientation, $R(3) \ t(3)$
Finding Camera Orientation and Internal Parameters

- Left 3x3 submatrix \( M \) of \( P \) is of form \( M=KR \)
  - \( K \) is an upper triangular matrix
  - \( R \) is an orthogonal matrix

- Any non-singular square matrix \( M \) can be decomposed into the product of an upper-triangular matrix \( K \) and an orthogonal matrix \( R \) using the RQ factorization.

\[
P = [KR \mid -KR\tilde{C}]
\]

The translation obtained

\[
t = K^{-1} m_4
\]
Calibration example

197 points were used

<table>
<thead>
<tr>
<th></th>
<th>$f_y$</th>
<th>$f_x/f_y$</th>
<th>skew</th>
<th>$x_0$</th>
<th>$y_0$</th>
<th>residual</th>
</tr>
</thead>
<tbody>
<tr>
<td>linear</td>
<td>1673.3</td>
<td>1.0063</td>
<td>1.39</td>
<td>379.96</td>
<td>305.78</td>
<td>0.365</td>
</tr>
<tr>
<td>iterative</td>
<td>1675.5</td>
<td>1.0063</td>
<td>1.43</td>
<td>379.79</td>
<td>305.25</td>
<td>0.364</td>
</tr>
</tbody>
</table>
Geometric interpretation of algebraic error

Fig. 7.2. The DLT algorithm minimizes the sum of squares of geometric distance $\Delta$ between the point $X_i$ and the point $X'_i$ mapping exactly onto $x_i$ and lying in the plane through $X_i$ parallel to the principal plane of the camera. A short calculation shows that $wd = f\Delta$.

$$\hat{w}_i = \pm \|\hat{p}^3\| \text{ depth}(X; P).$$

Thus, the value $\hat{w}_i$ may be interpreted as the depth of the point $X_i$ from the camera in the direction along the principal ray, provided the camera is normalized so that $\|\hat{p}^3\|^2 = p_{31}^2 + p_{32}^2 + p_{33}^2 = 1$. Referring to figure 7.2 one sees that $\hat{w}_i d(x_i, \hat{x}_i)$ is proportional to $fd(X', X)$, where $f$ is the focal length and $X'_i$ is a point mapping to $x_i$ and lying in a plane through $X_i$ parallel to the principal plane of the camera. Thus, the algebraic error being minimized is equal to $f \sum_i d(X_i, X'_i)^2$.

This is not $\hat{X}_i$ which may not be perpendicular to the principal axis. DLT biased toward minimizing focal length at the cost of slight increase of the 3D geometric error. This normalization leads to 3D minimization and is not affected by similarity transform.
Estimation of affine camera

\[
\begin{bmatrix}
0^T & -w_i X_i^T & y_i X_i^T \\
w_i X_i^T & 0^T & -x_i X_i^T
\end{bmatrix}
\begin{bmatrix}
P_1 \\
P_2 \\
P_3
\end{bmatrix} = 0
\]

\[
\begin{bmatrix}
0^T & -X_i^T \\
X_i^T & 0^T
\end{bmatrix}
\begin{bmatrix}
P_1 \\
P_2
\end{bmatrix} + \begin{bmatrix}
y_i \\
-x_i
\end{bmatrix} = 0
\]

\[
\|\Delta p\|^2 = \sum_i \left( x_i - P_1^T X_i \right)^2 + \left( y_i - P_2^T X_i \right)^2 = \sum_i d(x_i, \hat{x}_i)^2
\]

algebraic error = geometric error solved by least squares

Fig. 6.8. **Perspective vs weak perspective projection.** The action of the weak perspective camera is equivalent to orthographic projection onto a plane (at \( Z = d_0 \)), followed by perspective projection from the plane. The difference between the perspective and weak perspective image point depends both on the distance \( \Delta \) of the point \( X \) from the plane, and the distance of the point from the principal ray.
Restricted camera estimation

Find best fit that satisfies part of assumptions

- skew $s$ is zero
- pixels are square
- principal point is known
- complete camera matrix $K$ is known

$$K = \begin{bmatrix}
\alpha_x & s & x_0 \\
\alpha_y & y_0 & 1
\end{bmatrix}$$

Minimize algebraic error

$\rightarrow$ parameters $q \rightarrow P = K[R|-RC]$  $p = g(q)$

$\rightarrow$ minimize $||Ag(q)||$

Minimize geometric error (Levenberg-Marquardt)

$\rightarrow$ impose constraint through parametrization (example: $x_0$, $y_0$, $\alpha$)

9 para. unknown

error in the image  $\rightarrow \mathbb{R}^{9} \rightarrow \mathbb{R}^{2n}$

error in 3D and 2D  $\rightarrow \mathbb{R}^{3n+9} \rightarrow \mathbb{R}^{5n}$

estimates 2D, 3D
Reduced measurement matrix

One only has to work with 12x12 matrix, not 2nx12

\[ \|A_p\|_2 = p^T A^T A p = \|\hat{A} p\|_2 \]

\[ A^T A = (VDU^T)(UDV^T) = (VD)(DV^T) = \hat{A}^T \hat{A} \]

Another way of obtaining \( \hat{A} \) is to use the QR decomposition \( A = Q \hat{A} \), \( Q \) has orthogonal columns and \( \hat{A} \) is upper triangular and square.

mapping \( q \mapsto \hat{A}g(q) \) is a mapping from \( \mathbb{R}^9 \) to \( \mathbb{R}^{12} \)

\[ P = K \begin{bmatrix} R & -R \hat{C} \end{bmatrix} \]

\( P \) satisfies the condition \( p_{31}^2 + p_{32}^2 + p_{33}^2 = 1 \)

the last row of \( R \) is the same like \( p_3 \)-s above

<table>
<thead>
<tr>
<th></th>
<th>( f_y )</th>
<th>( f_x/f_y )</th>
<th>skew</th>
<th>( x_0 )</th>
<th>( y_0 )</th>
<th>residual</th>
</tr>
</thead>
<tbody>
<tr>
<td>algebraic</td>
<td>1633.4</td>
<td>1.0</td>
<td>0.0</td>
<td>371.21</td>
<td>293.63</td>
<td>0.601</td>
</tr>
<tr>
<td>geometric</td>
<td>1637.2</td>
<td>1.0</td>
<td>0.0</td>
<td>371.32</td>
<td>293.69</td>
<td>0.601</td>
</tr>
</tbody>
</table>

Table 7.2. Calibration for a restricted camera matrix.
CALIBRATION COVARIANCE ESTIMATION

$$\Sigma_{camera} = (J_{f|P} \Sigma_{points}^{-1} J_{f|P})^+$$

$J_{f|P}$ is the Jacobian matrix of the measured points in the terms of the camera parameters and $\Sigma_{points}$ the uncertainty of the measured points.

Uncertainty of the 2D and 3D points obtained at the end is

$$\hat{\Sigma}_{points} = J_{f|P} \Sigma_{camera} J_{f|P}^T$$

Assume $3 \times 3$ covariance extracted for the camera center $\Sigma_C$. The confidence ellipsoid of the camera center

$$(C - \bar{C})^T \Sigma_C^{-1} (C - \bar{C}) = k^2$$

where $k^2$ is a close to one probability value, say 0.95.

The $Z$ coordinate in 3D is much more sensitive to errors than $X$, $Y$. 
Fig. 7.3. **Camera centre covariance ellipsoids.** (a) Five images of Stanislas square (Nancy, France), for which 3D calibration points are known. (b) Camera centre covariance ellipsoids corresponding to each image, computed for cameras estimated from the imaged calibration points. Note, the typical cigar shape of the ellipsoid aligned towards the scene data. Figure courtesy of Vincent Lepetit, Marie-Odile Berger and Gilles Simon.
Calibration matrix $\mathbf{K}$

\[ \mathbf{x} = \mathbf{K}[\mathbf{I} \mid 0][\mathbf{d}] \]

\[ \mathbf{d} = \mathbf{K}^{-1}\mathbf{x} \]

in camera coor. direction $\mathbf{d}$ projects to point $\mathbf{x}$

\[ \cos \theta = \frac{\mathbf{d}_1^T \mathbf{d}_2}{\sqrt{(\mathbf{d}_1^T \mathbf{d}_1)(\mathbf{d}_2^T \mathbf{d}_2)}} = \frac{\mathbf{x}_1^T (\mathbf{K}^{-T} \mathbf{K}^{-1}) \mathbf{x}_2}{\sqrt{(\mathbf{x}_1^T (\mathbf{K}^{-T} \mathbf{K}^{-1}) \mathbf{x}_1)(\mathbf{x}_2^T (\mathbf{K}^{-T} \mathbf{K}^{-1}) \mathbf{x}_2)}} \]

**Result 8.16.** An image line $\mathbf{l}$ defines a plane through the camera centre with normal direction $\mathbf{n} = \mathbf{K}^T \mathbf{l}$ measured in the camera’s Euclidean coordinate frame.

\[ \mathbf{d}^T \mathbf{n} = \mathbf{x}^T \mathbf{K}^{-T} \mathbf{n} = 0 \]

\[ \mathbf{x}^T \mathbf{l} = 0, \text{ it follows that } \mathbf{l} = \mathbf{K}^{-T} \mathbf{n} \]

$\mathbf{K}$ known, the camera Euclidean frame is *calibrated*. 
Properties of image of the absolute conic (IAC)

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

mapping between \( \pi_\infty \) to an image is given by the planar homography \( x = Hd \), with \( H = KR \)

\[ \omega = \left( KK^T \right)^{-1} = K^{-T}K^{-1} \quad \omega^* = \omega^{-1} = KK^T \]

-- IAC depends only internal parameters

\[ \omega^* = P\Omega_\infty^*P^T = KR \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} I \\ 0 \end{bmatrix} R^T K^T = KK^T \]

\[ \omega = (KK^T)^{-1} = K^{-T}K^{-1} \quad \text{a point conic - IAC} \]

Like \( \Omega_\infty \) the conic \( \omega \) is an imaginary point conic with no real points.
Proof. From result 2.13(p37) under a point homography $x \mapsto Hx$ a conic $C$ maps as $C \mapsto H^{-T}CH^{-1}$. It follows that $\Omega_\infty$, which is the conic $C = \Omega_\infty = \mathbb{I}$ on $\pi_\infty$, maps to $\omega = (KR)^{-T}\mathbb{I}(KR)^{-1} = K^{-T}R R^{-1}K^{-1} = (KK^T)^{-1}$. So the IAC $\omega = (KK^T)^{-1}$. \hfill \Box

-- The angle between two rays is invariant when $x' = Hx$. The IAC transforms $H^{-T}\omega H^{-1}$

-- perpendicular direction in the scene (point in the image) or lines (planes in the scene)

\begin{align*}
\cos \theta &= \frac{x_1^T \omega x_2}{\sqrt{x_1^T \omega x_1} \sqrt{x_2^T \omega x_2}} \\

v_1^T \omega v_2 &= 0 \\

l_1^T \omega^* l_2 &= 0
\end{align*}

-- then $K$ is also determined. This follows because a symmetric matrix $\omega$ may be uniquely decomposed into a product $\omega^* = KK^T$ of an upper-triangular matrix with positive diagonal entries and its transpose by the Cholesky factorization (see result A4.5(p582)).

-- It was seen in chapter 3 that a plane $\pi$ intersects $\pi_\infty$ in a line, and this line intersects $\Omega_\infty$ in two points which are the circular points of $\pi$. The imaged circular points lie on $\omega$ at the points at which the vanishing line of the plane $\pi$ intersects $\omega$.

The final two properties of $\omega$ are the basis for a calibration algorithm.
example: angle between two vanishing points

\[ \cos(d_1, d_2) \]

\[ \text{cos} (d_1, d_2) \]

\[ \cos \theta = \frac{V_1^T \omega \ V_2}{\sqrt{V_1^T \omega \ V_1} \sqrt{V_2^T \omega \ V_2}} \]

If between two rays in 3D

\[ \theta = 90 \rightarrow V_1^T \omega \ V_2 = 0 \]
Calibration of $K$ in a single view: skew=0, square pixels

Compute $\omega$:

6 unknown - 1 scale
5 constraints $\Rightarrow$
$\omega$ known up to scale

$$\omega = \begin{bmatrix} \omega_1 & \omega_2 & \omega_4 \\ \omega_2 & \omega_3 & \omega_5 \\ \omega_4 & \omega_5 & \omega_6 \end{bmatrix}$$

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

(Cholesky factorization)
an example:

\[ \theta = 90 \]

\[
\begin{cases}
    v_1^T \omega v_3 = 0 \\
    \omega = (K K^T)^{-1}
\end{cases}
\]

a constraint on \( K \)
A simple calibration device with planes
not orthogonal planes!

(i) compute $H$ for each square, \textit{separately}
corners $(0,0),(1,0),(0,1),(1,1)$ in the world

(ii) compute the imaged circular points $H(1,\pm i,0)^T$
for a plane

(iii) fit a conic with 5 unknown $< 2 \times 3$ equations

(iv) compute $K$ from $\omega$ through Cholesky factorization

\[
K = \begin{bmatrix}
1108.3 & -2.8 & 525.8 \\
0 & 1097.8 & 395.9 \\
0 & 0 & 1
\end{bmatrix}
\]
Calibration Procedure

see webpage

Camera Calibration Toolbox for Matlab

http://www.vision.caltech.edu/bouguetj/calib_doc/index.html#examples

with planes
Take images of a planar checkerboard.
For each image click in this order on the four extreme corners.

Click #1 (origin)

Click #2

Click #3

Click #4
The image corners are then automatically extracted.
Radial distortion included in the 2D calibration.
If you do a recomputation of corners and estimate again, the reprojection error will decrease an order of magnitude more.
camera viewpoint
world viewpoints - 20 images
After the camera was calibrated...

\[
P = K[R \quad T]
\]

If \( K, R, t \) are known – the center \( C \) is also defined. Can we estimate \( P \) from \( p \) (2D) based on a single image?

**NO** - 😞 since \( P \) can be anywhere along the line defined by \( C \) and \( p \).
Orthogonality through pole-polar relationships in plane at infinity

\[ \cos \theta = \frac{d_1^T \Omega_{\infty} d_2}{\sqrt{d_1^T \Omega_{\infty} d_1} \sqrt{d_2^T \Omega_{\infty} d_2}} \]

\(d_1\) and \(d_2\) are the directions of the lines.

\[ x_1^T \omega x_2 = 0 \]

**Fig. 8.13.** Orthogonality represented by conjugacy and pole–polar relationships. (a) Image points \(x_1, x_2\) back-project to orthogonal rays if the points are conjugate with respect to \(\omega\), i.e. \(x_1^T \omega x_2 = 0\). (b) The point \(x\) and line \(l\) back-project to a ray and plane that are orthogonal if \(x\) and \(l\) are pole–polar with respect to \(\omega\), i.e. \(l = \omega x\). For example (see section 8.6.3), the vanishing point of the normal direction to a plane and the vanishing line of the plane are pole–polar with respect to \(\omega\).

\[ l = \omega \]
\[ K n = K^{\{-T\}} n \]
\[ n \text{ the normal to the plane } PI \]
\[ n = K^{\top} l \]
Vanishing points

*Calibrated camera:* rotation (no translation) computed from two vanishing points. $v_i \rightarrow v'_i$

unit normal

$$d_i = k^{-1}v_i/\|k^{-1}v_i\| \quad d'_i = Rd_i$$

two vanishing pairs only $\rightarrow R$

The lines go through a single vanishing point.

---

Fig. 8.15. ML estimate of a vanishing point from imaged parallel scene lines. (a) Estimating the vanishing point $v$ involves fitting a line (shown thin here) through $v$ to each measured line (shown thick here). The ML estimate of $v$ is the point which minimizes the sum of squared orthogonal distances between the fitted lines and the measured lines’ end points. (b) Measured line segments are shown in white, and fitted lines in black. (c) A close-up of the dashed square in (b). Note the very slight angle between the measured and fitted lines.
Vanishing lines

angle between two nonparallel scene planes, intersects \( l_1 \) and \( l_2 \) the two vanishing lines in the image

two vanishing points:
--a vanishing point is the walls,
--other vanishing point is the perpendicular lines on ground,
gives the vanishing line
example: equally spaced parallel lines in 3D

parallel lines $ax' + by' + n = 0$ $n=0,1,2,...$ written as $l'_n = (a,b,n)^T = (a, b, 0)^T + n(0, 0, 1)^T$

$(0, 0, 1)^T$ line at infinity of scene plane $x = H x'$ gives the line in the image $l_n = H^{\{T\}} l'_n = l_0 + nl$ where $l$ is the image of $(0, 0, 1)^T$

the vanishing line of the plane $(n=0,1,2)$

$$l = (l_0 \times l_2)^T (l_1 \times l_2) l_1 + 2 (l_0 \times l_1)^T (l_2 \times l_1) l_2$$

$l = s_1 l_1 + s_2 l_2$ use $n=1,2$ to find $s_{1,2}$ three equations
Orthogonality relationships

The vanishing points of lines with perpendicular directions satisfy conjugates
\[ \mathbf{v}_1^T \omega \mathbf{v}_2 = 0. \quad (8.16) \]

If a line is perpendicular to a plane then their respective vanishing point \( \mathbf{v} \) and vanishing line \( \mathbf{l} \) are related by pole - polar relation
\[ \mathbf{l} = \omega \mathbf{v} \quad (8.17) \]

and inversely \( \mathbf{v} = \omega^* \mathbf{l} \).

The vanishing lines of two perpendicular planes satisfy \( \mathbf{l}_1^T \omega^* \mathbf{l}_2 = 0 \).
Affine 3D measurements and reconstruction

Criminisi '99 (PhD)

Given in the image a scene plane's *vanishing line* and a *vanishing point* for a direction *not parallel to the plane*, (will take the perpendicular direction to the plane) ==> *affine 3D properties* can be computed for the scene. The method does *not* require internal calibration, $K$.

The *horizon* (horizontal ground plane) will be the vanishing line with the vanishing point orthogonal to the plane.

**Result 8.24.** Given the vanishing line of the ground plane $l$ and the vertical vanishing point $v$, then the relative length of vertical line segments can be measured provided their end point lies on the ground plane.
Fig. 8.20. **Computing length ratios of parallel scene lines.** (a) **3D geometry:** The vertical line segments $L_1 = (B_1, T_1)$ and $L_2 = (B_2, T_2)$ have length $d_1$ and $d_2$ respectively. The base points $B_1, B_2$ are on the ground plane. We wish to compute the scene length ratio $d_1 : d_2$ from the imaged configuration. (b) In the scene the length of the line segment $L_1$ may be transferred to $L_2$ by constructing a line parallel to the ground plane to generate the point $\tilde{T}_1$. (c) **Image geometry:** $l$ is the ground plane vanishing line, and $v$ the vertical vanishing point. A corresponding parallel line construction in the image requires first determining the vanishing point $u$ from the images $b_i$ of $B_i$, and then determining $\hat{t}_1$ (the image of $\tilde{T}_1$) by the intersection of $l_2$ and the line $(\hat{t}_1, u)$. (d) The line $l_3$ is parallel to $l_1$ in the image. The points $\hat{t}_1$ and $\hat{t}_2$ are constructed by intersecting $l_3$ with the lines $(\hat{t}_1, \hat{t}_1)$ and $(\hat{t}_1, \hat{t}_2)$ respectively. The distance ratio $d(b_2, \hat{t}_1) : d(b_2, \hat{t}_2)$ is the computed estimate of $d_1 : d_2$. 
Objective
Given the vanishing line of the ground plane \( l \) and the vertical vanishing point \( v \) and the top \((t_1, t_2)\) and base \((b_1, b_2)\) points of two line segments as in figure 8.20, compute the ratio of lengths of the line segments in the scene.

Algorithm

(i) Compute the vanishing point \( u = (b_1 \times b_2) \times l \).

(ii) Compute the transferred point \( \tilde{t}_1 = (t_1 \times u) \times l_2 \) (where \( l_2 = v \times b_2 \)).

(iii) Represent the four points \( b_2, \tilde{t}_1, t_2 \) and \( v \) on the image line \( l_2 \) by their distance from \( b_2 \), as \( 0, \tilde{t}_1, t_2 \) and \( v \) respectively.

(iv) Compute a 1D projective transformation \( H_{2 \times 2} \) mapping homogeneous coordinates \((0, 1) \mapsto (0, 1)\) and \((v, 1) \mapsto (1, 0)\) (which maps the vanishing point \( v \) to infinity). A suitable matrix is given by

\[
H_{2 \times 2} = \begin{bmatrix}
1 & 0 \\
1 & -v
\end{bmatrix}.
\]

(v) The (scaled) distance of the scene points \( \widetilde{T}_1 \) and \( T_2 \) from \( B_2 \) on \( L_2 \) may then be obtained from the position of the points \( H_{2 \times 2}(t_1, 1)^T \) and \( H_{2 \times 2}(t_2, 1)^T \). Their distance ratio is then given by

\[
\frac{d_1}{d_2} = \frac{\tilde{t}_1(v - t_2)}{t_2(v - \tilde{t}_1)}
\]

Measuring a person’s height in a single image

Fig. 8.21. Height measurements using affine properties. (a) The original image. We wish to measure the height of the two people. (b) The image after radial distortion correction (see section 7.4). (c) The vanishing line (shown) is computed from two vanishing points corresponding to horizontal directions. The lines used to compute the vertical vanishing points are also shown. The vertical vanishing point is not shown since it lies well below the image. (d) Using the known height of the filing cabinet on the left of the image, the absolute height of the two people are measured as described in algorithm 8.1. The measured heights are within 2 cm of ground truth. The computation of the uncertainty is described in [Criminisi-00].
two references reduce uncertainty

reference height is in the middle (Christ)
Determining $K$ from a single image

$\omega$ symmetric, can be less than 6-1=5 unknowns:
- **zero skew** -- the two vanishing points are perpendicular
- **square pixels** -- $K_{11} = K_{22}$

Algorithm to find $K$ after finding $\omega$:

<table>
<thead>
<tr>
<th>Condition</th>
<th>constraint</th>
<th>type</th>
<th># constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>vanishing points $v_1, v_2$ corresponding to orthogonal lines</td>
<td>$v_1^T \omega v_2 = 0$</td>
<td>linear</td>
<td>1</td>
</tr>
<tr>
<td>vanishing point $v$ and vanishing line $l$ corresponding to orthogonal line and plane</td>
<td>$[l] \times \omega v = 0$</td>
<td>linear</td>
<td>2</td>
</tr>
<tr>
<td>metric plane imaged with known homography $H = [h_1, h_2, h_3]$</td>
<td>$h_1^T \omega h_2 = 0$; $h_1^T \omega h_1 = h_2^T \omega h_2$</td>
<td>linear</td>
<td>2</td>
</tr>
<tr>
<td>zero skew</td>
<td>$\omega_{12} = \omega_{21} = 0$</td>
<td>linear</td>
<td>1</td>
</tr>
<tr>
<td>square pixels</td>
<td>$\omega_{12} = \omega_{21} = 0$; $\omega_{11} = \omega_{22}$</td>
<td>linear</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 8.1. Scene and internal constraints on $\omega$. 

$H = K [r_1 \; r_2 \; t]$ 
$Z = 0$!
Objective

Compute $K$ via $\omega$ by combining scene and internal constraints.

Algorithm

(i) Represent $\omega$ as a homogeneous 6-vector $\mathbf{w} = (w_1, w_2, w_3, w_4, w_5, w_6)^T$ where:

$$
\omega = \begin{bmatrix}
w_1 & w_2 & w_4 \\
w_2 & w_3 & w_5 \\
w_4 & w_5 & w_6 
\end{bmatrix}
$$

(ii) Each available constraint from table 8.1 may be written as $\mathbf{a}^T \mathbf{w} = 0$. For example, for the orthogonality constraint $\mathbf{u}^T \mathbf{w} \mathbf{v} = 0$, where $\mathbf{u} = (u_1, u_2, u_3)^T$ and $\mathbf{v} = (v_1, v_2, v_3)^T$, the 6-vector $\mathbf{a}$ is given by

$$
\mathbf{a} = (v_1 u_1, v_1 u_2 + v_2 u_1, v_2 u_2, v_1 u_3 + v_3 u_1, v_2 u_3 + v_3 u_2, v_3 u_3)^T.
$$

Similar constraints vectors are obtained from the other sources of scene and internal constraints. For example a metric plane generates two such constraints.

(iii) Stack the equations $\mathbf{a}^T \mathbf{w} = 0$ from each constraint in the form $\mathbf{A} \mathbf{w} = \mathbf{0}$, where $\mathbf{A}$ is an $n \times 6$ matrix for $n$ constraints.

(iv) Solve for $\mathbf{w}$ using the SVD as in algorithm 4.2(p109). This determines $\omega$.

(v) Decompose $\omega$ into $K$ using matrix inversion and Cholesky factorization (see section A4.2.1(p582)).

Algorithm 8.2. Computing $K$ from scene and internal constraints.
Calibration from three orthogonal vanishing points

Zero skew and square pixels. Three unknowns.

$$\omega = \begin{bmatrix} w_1 & 0 & w_2 \\ 0 & w_1 & w_3 \\ w_2 & w_3 & w_4 \end{bmatrix}.$$  \hspace{1cm} \text{A 3x4 matrix} \hspace{1cm} A \omega = 0

Each pair of vanishing points $v_i, v_j$ generates an equation $v_i^T \omega v_j = 0$.

Fig. 8.22. For the case that image skew is zero and the aspect ratio unity the principal point is the orthocentre of an orthogonal triad of vanishing points. (a) Original image. (b) Three sets of parallel lines in the scene, with each set having direction orthogonal to the others. (c) The principal point is the orthocentre of the triangle with the vanishing points as vertices.

How to find the principal point and the focal distance?
\( \mathbf{n} \) is normal on the plane intersection image plane, in \( \mathbf{v}_3 \)
\( \mathbf{x} \) on \( \mathbf{l}_3 \) is on the perpendicular line from \( \mathbf{v}_3 \)
The principal point is the orthocenter.
\[ \alpha /d(p,x) = d(p,v_3) /\alpha \]  

geometric mean theorem

applied to \( \text{apx, Cpx} \)

\( C_p \) is alpha, from similar triangles \( \text{pa} \) is also alpha.

If a vanishing points is at infinity, the rank of \( \text{A} \) is only 2.

For \( \text{v}_3 \text{ v.p. at inf.} \) the orthocenter is on \( \text{l}_3 \), but \( x \) not defined.
Focal length is only unknown

Principal point in the center of the image. skew=0. Square pixels.

\( \omega = \text{diag}(1/f^2, 1/f^2, 1) \)

Two orthogonal vanishing point give the one constraint.

Vanishing line of facade \( l \): windows -- horizontal edges, pavement; vertical edges.

Given \( K \) and \( l \): synthetic rotations.

Here instead of the aspect ratio of a rectangle, use the vanishing line \( l \) and \( K \) to rotate in the image plane. Rotation is around the z-axis.

---

**Fig. 8.25. Plane rectification via partial internal parameters** (a) Original image. (b) Rectification assuming the camera has square pixels and principal point at the centre of the image. The focal length is computed from the single orthogonal vanishing point pair. The aspect ratio of a window in the rectified image differs from the ground truth value by 3.7%. Note that the two parallel planes, the upper building facade and the lower shopfront, are both mapped to fronto-parallel planes.
Camera rotation

\[ x = K[I | 0]X \]

\[ x' = K[R | 0]X = KRK^{-1}x = Hx \]

\[ \{\mu, \mu e^{i\theta}, \mu e^{-i\theta}\} \]

\[ \det H = 1, \text{ then } \mu = 1 \]

conjugate rotation A.7.1

\[ H = KRK^{-1} \text{ has an eigenvector } Ka \]

\[ HKa = KRa = Ka \]

rotation \( \sim y\)-axis

a eigenvector of \( R \)

\[ v = Ka \text{ vanishing point of the rotation axis} \]

translation too
Planar homography mosaicing

one image is the reference

compute all $H$ relative to reference

projectively wrap, augment reference

with non-overlapping parts. Repeat till the end.
Slightly other example:

Fig. 8.9. Planar panoramic mosaicing. Eight images (out of thirty) acquired by rotating a camcorder about its centre. The thirty images are registered (automatically) using planar homographies and composed into the single panoramic mosaic shown. Note the characteristic “bow tie” shape resulting from registering to an image at the middle of the sequence.
Image warping with homographies

homography so that image is parallel to floor

The trapezoid goes to rectangle.
homography so that image is parallel to floor

black area where no pixel maps to

image plane below
homography so that image is parallel to floor

homography so that image is parallel to right wall

black area where no pixel maps to
Measurements on planes

Approach: unwarp the measured by the distance between the squares. Four point gives the homography.

for the synthetic camera views take measurement on the image
Measurements on the corridor wall.
Calibrating conic

Image of absolute conic (IAC) = imaginary conic (not visible)

\[ P = K[I \mid 0] \]

Calibrating conic: apex angle 45 deg., axis the principal axis of the camera. Independent of the position of the camera.

The cone \( X^2 + Y^2 = Z^2 \) and maps to the image

\[
C = K^{-T} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \end{bmatrix} K^{-1}
\]
Orthogonality and the calibrating conic

Two points $x, x'$ are perpendicular when $x'^T \omega x = 0$ and $x'$ lying on the line $\omega x$.

$C = K^{-T}DK^{-1}$, where $D = \text{diag}(1, 1, -1)$, we find

$C = (K^{-T}K^{-1})(KDK^{-1}) = \omega S$

where $S = KDK^{-1}$. However, for any point $x$, the product $Sx$ is the reflection of the point through the center of the conic $C$. ($\det D = -1$)

$x'^T \omega x = x'^T C x$

Fig. 8.28. To construct the line perpendicular to the ray through image point $x$ proceed as follows: (i) Reflect $x$ through the centre of $C$ to get point $\hat{x}$ (i.e. at the same distance from the centre as $x$). (ii) The desired line is the polar of $\hat{x}$. 
Calibrating conic given by three orthogonal vanishing points

Find the orthocenter of the triangle from the three v.p. Reflect one of the vanishing points \( \mathbf{v}_1 \) to get \( \bar{\mathbf{v}}_1 \). The radius of conic \( C \) is determined that the polar of \( \bar{\mathbf{v}}_1 \). The line passes through \( \mathbf{v}_2 \) and \( \mathbf{v}_3 \).

\[
\ell(\mathbf{v}_2, \mathbf{v}_3) = C \bar{\mathbf{v}}_1
\]

Fig. 8.29. The calibrating conic computed from three orthogonal vanishing points. (a) The geometric construction. (b) The calibrating conic for the image of figure 8.22.
The importance of the camera center

two cameras

$$P = KR[I \mid -\tilde{C}], P' = K'R'[I \mid -\tilde{C}]$$

$$P' = K'R'(KR)^{-1} P$$

same camera center

$$x' = P'X = K'R'(KR)^{-1} PX = K'R'(KR)^{-1} x$$

planar homography

$$x' = Hx \text{ with } H = K'R'(KR)^{-1}$$
Moving the image plane by focal length zooming

\[
x = K[I \mid 0]X
\]
\[
x' = K'[I \mid 0]X = K'(K)^{-1}x
\]
\[
x' = Hx
\]

\(k\) is to move the image point \(\tilde{x}\) on a line radiating from the principal point \(\tilde{x}_0\)

\[
\tilde{x}' = k\tilde{x} + (1-k)\tilde{x}_0
\]

\[
\begin{bmatrix}
kA & \tilde{x}_0 \\
0 & 1
\end{bmatrix}
= K'
\]
\[
H = K'(K)^{-1} = \begin{bmatrix}
kI & (1-k)\tilde{x}_0 \\
0^T & 1
\end{bmatrix}
\]

Zooming with \(k\) is to multiply \(K\) with \(\text{diag}(k, k, 1)\) from the right.
Projective (reduced) notation

The world coordinates chosen as the three vanishing points and origin. For camera coordinates take the three axes and the middle point.

\[
\begin{align*}
\mathbf{X}_1 &= (1,0,0,0)^T, \quad \mathbf{X}_2 = (0,1,0,0)^T, \quad \mathbf{X}_3 = (0,0,1,0)^T, \quad \mathbf{X}_4 = (0,0,0,1)^T \\
\mathbf{x}_1 &= (1,0,0)^T, \quad \mathbf{x}_2 = (0,1,0)^T, \quad \mathbf{x}_3 = (0,0,1)^T, \quad \mathbf{x}_4 = (1,1,1)^T
\end{align*}
\]

\[
\mathbf{P} = \begin{bmatrix}
a & 0 & 0 & -d \\
0 & b & 0 & -d \\
0 & 0 & c & -d \\
\end{bmatrix} \quad \mathbf{x} = \mathbf{PX}
\]

\[
\mathbf{C} = (a^{-1}, b^{-1}, c^{-1}, d^{-1})^T
\]

the center, because \(\mathbf{PC} = 0\)

Specify by three DOF of the center.

Multiply \(\mathbf{P}\) from the left with 2D homography 3+8=11 DOF

Projective equivalent cameras.
Moving the camera center

motion parallax between $x'_1$ and $x'_2$

Fig. 8.10. **Motion parallax.** The images of the space points $X_1$ and $X_2$ are coincident when viewed by the camera with centre $C$. However, when viewed by a camera with centre $C'$, which does not lie on the line $L$ through $X_1$ and $X_2$, the images of the space points are not coincident. In fact the line through the image points $x'_1$ and $x'_2$ is the image of the ray $L$, and will be seen in chapter 9 to be an **epipolar line.** The vector between the points $x'_1$ and $x'_2$ is the parallax.