AUTO - CALIBRATION
Auto- (self-) calibration computes the metric (including internal camera properties) from multiple projective, uncalibrated images.

Depending on what constraints are given, we need less or more images to eliminate the projective ambiguity:

projective reconstruction \( \{P^i, X_j\} \)

determine a rectifying

4x4 homography \( H \)

to have the metric reconstruction.

Will study two methods.

*First* defines \( H \) directly from the absolute dual quadric, the *second* uses the stratified solution.

The second method has the *affine* to *metric* part only a linear estimation.
Algebraic framework

Given $m$ cameras, we want to determine in a Euclidean world frame where a common $H$ (4x4) is to be find.

First camera has no rotation and translation.

$$P_M^i = P^iH \quad i = 1, \ldots, m$$

where

$$P_M^1 = K^1[I \mid 0]$$

where the scale of $H$ is one, and is based on the first camera.

The vector $v$ and 3x3 matrix $K^{^1}$ specifies the plane at infinity

$$\pi_\infty = H^{-T} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} (K^1)^{-T} \\ -(K^1)^{-T}v \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -(K^1)^{-T}v \\ 1 \end{pmatrix}$$

First camera, upper-triangular, $K=K^{^1}$ matrix.

Conversely, if the first camera is known, the matrix $H$ is determined.
Has to determine $3+5=8 \ (p + K^1)$ parameters for $H$ which is equivalent with plane at infinity (3) and absolute conic (5).

All the cameras projective reconstruction

multiplies $P^i$ with $H$

and gives the form

$$K^iR^i = (A^i - a^ip^T)K^1 \quad \text{for } i = 2, \ldots, m$$

$$K^iK^{iT} = (A^i - a^ip^T)K^1K^{1T}(A^i - a^ip^T)^T$$

```
\omega^i = (A^i - a^ip^T) \omega^1 (A^i - a^ip^T)^T
```

Relate the unknown $\omega$-s, $i=1, \ldots, m$, and $p$, to known $A^i$, $a^i$.

Example. *Same internal parameters.*

Each $i=2, \ldots, m$ has 5 constraints since $KK^T$ is symmetric. If independent for each view, $5(m-1) \geq 8$. Three views is enough.
Absolute dual quadric calibration

Degenerate \textit{rank} 3, symmetric, positive semi-definite 4x4 matrix encoding both the plane at infinity and the absolute conic.

$$\omega^* = PQ^*P^T \quad Q^*_\infty \pi_\infty = 0$$

Determinant equal zero.

In a Euclidean frame $Q^*_\infty$ has the canonical form

$$\tilde{I} = \begin{bmatrix} I_{3 \times 3} & 0 \\ 0^T & 0 \end{bmatrix}.$$

In a projective coordinate frame $Q^*_\infty$ has the form $Q^*_\infty = H\tilde{I}H^T$

From projective to Euclidean a 3D point multiplied by $H^{-1}$ since the cameras are multiplied by $H \Rightarrow 8$ parameters.

$$Q^*_\infty = H\tilde{I}H^T = \begin{bmatrix} K^1K^1^T & -K^1K^1^Tp \\ -p^TK^1K^1^T & p^TK^1K^1^Tp \end{bmatrix} = \begin{bmatrix} \omega^*1 & -\omega^*1p \\ -p^T\omega^*1 & p^T\omega^*1p \end{bmatrix}$$

obtaining the previous result

$$P^i = [A^i \mid a^i]$$

$$\omega^{*i} = P^iQ^*_\infty P^{i\top} = (A^i - a^i p^T)\omega^*1 (A^i - a^i p^T)^T$$
Number of parameters to perform calibration: 8. $m$ views; $k$ internal parameters known; $f$ fixed but unknown, $k + f \leq 5$.

$$mk + (m-1)f \geq 8$$
gives $m \implies$ how many views

example: $2m + 0.(m-1) \geq 8\text{ thus } m = 4$

<table>
<thead>
<tr>
<th>Condition</th>
<th>fixed $f$</th>
<th>known $k$</th>
<th>views $m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant internal parameters</td>
<td>5</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Aspect ratio and skew known, focal length and principal point vary</td>
<td>0</td>
<td>2</td>
<td>4*</td>
</tr>
<tr>
<td>Aspect ratio and skew constant, focal length and principal point vary</td>
<td>2</td>
<td>0</td>
<td>5*</td>
</tr>
<tr>
<td>Skew zero, all other parameters vary</td>
<td>0</td>
<td>1</td>
<td>8*</td>
</tr>
<tr>
<td>p.p. known all other parameters vary</td>
<td>0</td>
<td>2</td>
<td>4*, 5(linear)</td>
</tr>
<tr>
<td>p.p. known skew zero</td>
<td>0</td>
<td>3</td>
<td>3(linear)</td>
</tr>
<tr>
<td>p.p., skew and aspect ratio known</td>
<td>0</td>
<td>4</td>
<td>2, 3(linear)</td>
</tr>
</tbody>
</table>

Table 19.3. The number of views $m$ required under various conditions in order for there to be enough constraints for auto-calibration. For those cases marked with an asterisk there may be multiple solutions, even for general motion between views.
The forms of \( \omega = (KK^T)^{-1} \) and \( \omega^* = \omega^{-1} = KK^T \) for a camera with calibration matrix \( K \) as in (6.10–p157) are

\[
\omega^* = \begin{bmatrix}
\frac{\alpha_x^2 + s^2 + x_0^2}{x_0} & \frac{s\alpha_y + x_0 y_0}{y_0} & x_0 \\
\frac{s\alpha_y + x_0 y_0}{x_0} & \frac{\alpha_y^2 + y_0^2}{y_0} & y_0 \\
x_0 & y_0 & 1
\end{bmatrix}
\] (19.9)

and

\[
\omega = \frac{1}{\alpha_x^2 \alpha_y^2} \begin{bmatrix}
\alpha_y^2 & -s\alpha_y & -x_0 \alpha_y + y_0 s\alpha_y \\
-s\alpha_y & \alpha_x^2 + s^2 & \alpha_y x_0 - \alpha_x^2 y_0 - s^2 y_0 \\
-x_0 \alpha_y + y_0 s\alpha_y & \alpha_y x_0 - \alpha_x^2 y_0 - s^2 y_0 & \alpha_x^2 \alpha_y^2 + \alpha_x^2 y_0^2 + (\alpha_y x_0 - s y_0)^2
\end{bmatrix}
\] (19.10)

If the skew is zero, i.e. \( s = 0 \), then the expressions simplify to

\[
\omega^* = \begin{bmatrix}
\frac{\alpha_x^2 + x_0^2}{x_0} & \frac{x_0 y_0}{y_0} & x_0 \\
x_0 y_0 & \frac{\alpha_y^2 + y_0^2}{y_0} & y_0 \\
x_0 & y_0 & 1
\end{bmatrix}
\] (19.11)

and

\[
\omega = \frac{1}{\alpha_x^2 \alpha_y^2} \begin{bmatrix}
\alpha_y^2 & 0 & -\alpha_y^2 x_0 \\
0 & \alpha_x^2 & -\alpha_x^2 y_0 \\
-\alpha_y^2 x_0 & -\alpha_x^2 y_0 & \alpha_y^2 \alpha_x^2 + \alpha_y^2 x_0^2 + \alpha_x^2 y_0^2
\end{bmatrix}
\] (19.12)

Table 19.1. The image of the absolute conic, \( \omega \), and dual image of the absolute conic, \( \omega^* \), written in terms of the camera internal parameters.

Linear constraints on \( \mathbf{Q}^*_\infty \) if the principal point is known...
...origin then can be in the principal point.

$$\omega^* = \begin{bmatrix} \alpha_x^2 + s^2 & s \alpha_y & 0 \\ s \alpha_y & \alpha_y^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

\begin{align*}
\omega_{13}^{*} &= \omega_{23}^{*} = 0 \\
(P_i^*Q_{\infty}^*P_i^T)_{13} &= 0 \text{ and } (P_i^*Q_{\infty}^*P_i^T)_{23} = 0
\end{align*}

<table>
<thead>
<tr>
<th>Condition</th>
<th>Constraint</th>
<th>Type</th>
<th># Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>zero skew</td>
<td>$\omega_{12}^<em>\omega_{33}^</em> = \omega_{13}^<em>\omega_{23}^</em>$</td>
<td>quadratic</td>
<td>$m$</td>
</tr>
<tr>
<td>principal point (p.p.) at origin</td>
<td>$\omega_{13}^* = \omega_{23}^* = 0$</td>
<td>linear</td>
<td>$2m$</td>
</tr>
<tr>
<td>zero skew (p.p. at origin)</td>
<td>$\omega_{12}^* = 0$</td>
<td>linear</td>
<td>$m$</td>
</tr>
<tr>
<td>fixed (unknown) aspect ratio (zero skew and p.p. at origin)</td>
<td>$\frac{\omega_{11}^<em>}{\omega_{22}^</em>} = \frac{\omega_{11}^<em>}{\omega_{22}^</em>}$</td>
<td>quadratic</td>
<td>$m - 1$</td>
</tr>
<tr>
<td>known aspect ratio $r = \alpha_y/\alpha_x$ (zero skew and p.p. at origin)</td>
<td>$r^2\omega_{11}^* = \omega_{22}^*$</td>
<td>linear</td>
<td>$m$</td>
</tr>
</tbody>
</table>

Table 19.2. **Auto-calibration constraints derived from the DIAC.** The number of constraints column gives the total number of constraints over $m$ views, assuming the constraint is true for each view. Each additional item of information generates additional equations. For example, if the principal point is known and skew is zero then there are 3 constraints per view.
Non-linear (quadratic) type relations in \( Q_\infty \).

*Constant internal parameters.*

\[
\omega^* = \omega^* \quad \text{for all } i \text{ and } j, \quad \text{which expands to } P^i Q^*_\infty P^i T = P^j Q^*_\infty P^j T
\]

\[
\omega_{11}^*/\omega_{11}^* = \omega_{12}^*/\omega_{12}^* = \omega_{13}^*/\omega_{13}^* = \omega_{22}^*/\omega_{22}^* = \omega_{23}^*/\omega_{23}^* = \omega_{33}^*/\omega_{33}^*
\]

Five equations between \( i \) and \( j \). The \( Q^*_\infty \), dual absolute quadric has 10 - scale - determinant = 8 unknown.

Three view enough, ten equations, 2/1 and 3/1.

*Zero skew.*

\[
\omega_{12}^* \omega_{33}^* = \omega_{13}^* \omega_{23}^*
\]

One constraint per view. Would need 8 views because

\[
\text{det } Q^*_\infty = 0
\]

and the scale can be factored out.
Symmetric, positive semi-definite matrix $Q^*$ eight parameters. Unknown elements of $Q^*_\infty$, and the unknown elements of each $\omega^*_i = K_i K_i^T$

Estimate with algebraic distance (unit Frobenius norms!) iteratively

$$\sum_i \|K_i K_i^T - P_i^T Q^*_\infty P_i^T\|_F^2$$

Exmp. focal distances only $f_i$, gives $m+3$ (p) eqn.

**Examples.** Just variable focal length.
Four linear constraints fixed in each view. **Two views** is enough assuming $\det Q^*_\infty = 0$ (non-linear) **Three views** solve uniquely but then determinant has to be made zero too!

The principal point is known.
Two linear constraints fixed in each view. **Four views** assuming the determinant already valid, **five views** and the determinant also applied.

$$\det Q^*_\infty = 0$$ has four solutions!!
Objective

Given a set of matched points across several views and constraints on the calibration matrices $K^i$, compute a metric reconstruction of the points and cameras.

Algorithm

(i) Compute a projective reconstruction from a set of views, resulting in camera matrices $P^i$ and points $X_j$.
(ii) Use (19.6) along with constraints on the form of the $\omega^{*i}$ arising from $K^i$ to estimate $Q^*_\infty$.
(iii) Decompose $Q^*_\infty$ as $H\tilde{H}H^T$, where $\tilde{H}$ is the matrix $\text{diag}(1, 1, 1, 0)$.
(iv) Apply $H^{-1}$ to the points and $H$ to the cameras to get a metric reconstruction.
(v) Use iterative least-squares minimization to improve the solution (see section 19.3.3).

Alternatively, the calibration matrix of each camera may be computed directly:

(i) Compute $\omega^{*i}$ for all $i$ using (19.6).
(ii) Compute the calibration matrix $K^i$ from the equation $\omega^* = KK^T$ by Cholesky factorization.


The positive definiteness of $\omega^*$ many times is not satisfied. Imposing positive semi-definiteness $Q^*_\text{inf}$ may not be successful in DIAC or... IAC, to be described.
Fig. 19.1. **Metric reconstruction for general motion.** (a)-(c) 3 views (of 5) acquired by a hand held camera. (d) and (e) Two views of a metric reconstruction computed from interest points matches over the five views. The cameras are represented by pyramids with apex at the computed camera centre. (f) and (g) two views of a texture mapped 3D model computed from the original images and reconstructed cameras using an area based stereo algorithm. Figures courtesy of Marc Pollefeys, Reinhard Koch, and Luc Van Gool.
Calibration based on stratification of geometry

- **Projective**: 15 DOF
- **Affine**: 12 DOF
  - plane at infinity parallelism
- **Metric**: 7 DOF
  - absolute conic angles, rel.dist.

Have to eliminate 15-7=8 parameters

More general

More structure
Removing the ambiguities: the Stratified reconstruction

• upgrade reconstruction from perspective to affine
  [by measuring the plane at infinity]

• upgrade reconstruction from affine to metric
  [by measuring the absolute conic]

Recovering the metric reconstruction from the perspective one is called self-calibration
Affine reconstruction - determining

Plane at infinity is known also as *modulus constraint*. The *internal parameters are constant* $K^i = K$, scale factor $\mu$ is included, but the superscripts on $R, A, a$ are not.

Infinite homography

$$A - ap^T = \mu KRK^{-1}$$

Eigenvalues of $(A - ap^T)$ are $\{\mu, \mu e^{i\theta}, \mu e^{-i\theta}\}$

$$\det(\lambda I - A + ap^T) = (\lambda - \lambda_1)(\lambda - \lambda_2)(\lambda - \lambda_3) = \lambda^3 - f_1\lambda^2 + f_2\lambda - f_3$$

$$f_1 = \lambda_1 + \lambda_2 + \lambda_3 = \mu(1 + 2\cos\theta)$$

$$f_2 = \lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3 = \mu^2(1 + 2\cos\theta)$$

$$f_3 = \lambda_1\lambda_2\lambda_3 = \mu^3.$$ 

Eliminate scalar $\mu$ and angle $\theta$ we obtain

$$f_3f_1^3 = f_2^3.$$
-- \( p \) appears rank-1 in \( ap^T \), linear terms in the \( f_{1,2,3} \). The relation is a quartic polynomial in \( p_{1,2,3} \). In each view are 4 solutions.

-- \( m=3 \) (intersection 3 quartic) determines \( p \). \( 4^3 \) solutions!

Additional, scene information has to be taken into account. Exmp. a vanishing line (two vanishing points) available in two views, \( \pi_\infty \) becomes a one-parameter ambiguity (vanishing pnt) and the quartic equation has only one variable.

Finding the plane at infinity is hardest in auto-calibration. If you have a vanishing point correspondence in \textit{two views} gives a point in \( \pi_\infty \). Three distant correspondences give the plane at infinity, as seen before.

\textit{Pure translation} between two views determines uniquely the plane at infinity. Epipolar constraint \( F=e'_x H_{\text{inf}} \).
Affine to metric - determining $\pi_\infty$

Infinite homography.

Plane at infinity $\pi_\infty = (p^T, 1)^T$ and camera matrices $[A^i \mid a^i]$

$$H_\infty^i = A^i - a^i p^T$$

If the first camera $[I \mid 0]$ can be converted.

Once the infinite homography is known, there is a linear relation between the omega-s. $K$ solved by Cholesky decomposition.

The scale factor being taken

$$\det H_\infty^i = 1$$

$$\omega_i^* = H_\infty^i \omega^* H_\infty^i \quad \text{T}$$

Same internal parameters...
...than the unknown conic has 6-1=5 unknown elements. Two views determine omega uniquely, as will see later.

Numerical stability is a big problem. Computation of $K$ from $H_\infty$ is extremely sensitive to accuracy of $H_\infty$ and then not always is possible to have a positive-definite matrix $\omega$ or similarly for $\omega^*$

Sensitivity is reduced if further motions, more $H_\infty$ are present.

Using IAC is preferred here. Zero-skew case is simpler and reflects better the calibration parameters. More linear constraints then DIAC.
Hard constraints, like square pixels, can be imposed directly to find a more reliable $K$ by adding $Cx=0$. The constrained total least squares is different... will see after the main algorithm.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Constraint</th>
<th>Type</th>
<th># Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>zero skew</td>
<td>$\omega_{12} = 0$</td>
<td>linear</td>
<td>$m$</td>
</tr>
<tr>
<td>principal point at origin</td>
<td>$\omega_{13} = \omega_{23} = 0$</td>
<td>linear</td>
<td>$2m$</td>
</tr>
<tr>
<td>known aspect ratio $r = \alpha_y/\alpha_x$</td>
<td>$\omega_{11} = r^2\omega_{22}$</td>
<td>linear</td>
<td>$m$</td>
</tr>
<tr>
<td>(assuming zero skew)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fixed (unknown) aspect ratio</td>
<td>$\omega_{11}^i/\omega_{22}^i = \omega_{11}^j/\omega_{22}^j$</td>
<td>quadratic</td>
<td>$m-1$</td>
</tr>
<tr>
<td>(assuming zero skew)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 19.4. Auto-calibration constraints derived from the IAC. These constraints are derived directly from the form of (19.10-p464) and (19.12-p464). The number of constraints column gives the total number of constraints over $m$ views, assuming the constraint is true for each view.
Objective

Given a projective reconstruction \( \{P^i, X_j\} \), where \( P^i = [A^i | a^i] \), determine a metric reconstruction via an intermediate affine reconstruction.

Algorithm

(i) **Affine rectification:** Determine the vector \( p \) that defines \( \pi_\infty \), using one of the methods described in section 19.5.1. At this point an affine reconstruction may be obtained as \( \{P^i H_p, H_p^{-1} X_j\} \) with

\[
H_p = \begin{bmatrix}
I & 0 \\
-p^T & 1
\end{bmatrix}.
\]

(ii) **Infinite homography:** Compute the infinite homography between the reference view and the others as

\[
H^i_\infty = (A^i - a^i p^T).
\]

Normalize the matrix so that \( \det H^i_\infty = 1 \).

(iii) **Compute \( \omega \):**
- In the case of constant calibration: rewrite the equations \( \omega = (H^i_\infty)^{-T} \omega (H^i_\infty)^{-1}, i = 1, \ldots, m \) as \( A c = 0 \) with \( A \) a \( 6m \times 6 \) matrix, and \( c \) the elements of the conic \( \omega \) arranged as a 6-vector, or
- For variable calibration parameters, use the equation \( \omega^i = (H^i_\infty)^{-T} \omega (H^i_\infty)^{-1} \) to express linear constraints on entries of \( \omega^i \) (e.g. zero skew) as linear equations in the entries of \( \omega \).

(iv) Obtain a least-squares solution to \( A c = 0 \) via SVD.

(v) **Metric rectification:** Determine the camera matrix \( K \) from the Cholesky decomposition \( \omega = (KK^T)^{-1} \). Then a metric reconstruction is obtained as \( \{P^i H_p H_A, (H_p H_A)^{-1} X_j\} \) with

\[
H_A = \begin{bmatrix}
K & 0 \\
0^T & 1
\end{bmatrix}.
\]

(vi) Use iterative least-squares minimization to improve the solution (see section 19.3.3).

**Algorithm 19.2.** Stratified auto-calibration algorithm using IAC constraints.
**TLS:** The exact constraints are described $\mathbf{C} \mathbf{x} = \mathbf{0}$. (qxn)

Extend with rows of $\mathbf{C}$ with zeros to a square matrix (nxn).

Find the $\mathbf{x}$ that minimizes $\|A\mathbf{x}\|$ subject to $\|\mathbf{x}\| = 1$ and $\mathbf{C} \mathbf{x} = \mathbf{0}$.

$x$ lie perpendicular to the rows of $\mathbf{C}$, in orthogonal complement of rows $\mathbf{C}$ ($\mathbf{n} \times \mathbf{r}$)($\mathbf{r} \times \mathbf{r}$)($\mathbf{r} \times \mathbf{n}$)

$\mathbf{C} \perp$ to be the matrix $\mathbf{V}$

$\mathbf{C} \mathbf{C} \perp = \mathbf{0}$

$\mathbf{C}$ and $\mathbf{C} \perp$ has orthogonal columns $\|\mathbf{x}\| = \|\mathbf{C} \perp \mathbf{x}'\| = \|\mathbf{x}'\|$

Find the $\mathbf{x}'$ that minimizes $\|A\mathbf{C} \perp \mathbf{x}'\|$ subject to $\|\mathbf{x}'\| = 1$.

\[ \mathbf{C} = \mathbf{U} \mathbf{D} \mathbf{V}^\top \]

with $\mathbf{r}$ non-zero diagonal entries $\mathbf{D}$

first $\mathbf{r}$ columns deleted ($\mathbf{n} \times \mathbf{n}-\mathbf{r}$)
Algorithm 5.4 is the *usual* total least squares procedure. Solution is the last column of $V$. 

---

**Objective**

Given an $m \times n$ matrix $A$ with $m \geq n$, find the vector $x$ that minimizes $\|Ax\|$ subject to $\|x\| = 1$ and $Cx = 0$.

**Algorithm**

(i) If $C$ has fewer rows than columns, then add zero-filled rows to $C$ to make it square. Compute the SVD $C = UDV^T$ where diagonal entries of $D$ are sorted with non-zero ones first. Let $C^\perp$ be the matrix obtained from $V$ by deleting the first $r$ columns of $V$, where $r$ is the number of non-zero entries in $D$ (the rank of $C$).

(ii) Find the solution to the minimization problem $AC^\perp x' = 0$ using algorithm A5.4. The solution is given by $x = C^\perp x'$.

---

Algorithm A5.5. Algorithm for constrained minimization.
Ambiguities with infinite homography

Rotation matrix $R$ has the unit eigenvector $v_r$.

The matrix $H^i_\infty = KR^i K^{-1}$ has the eigenvector $v_r = Kd_r$, where

$$\det H^i_\infty = 1$$

where the point $v_r$ is the vanishing point of the rotation axis.

$$\omega^* = H^i_\infty \omega^* H^i_\infty^T$$ always satisfied by a one-parameter family

$$\omega^*(\mu) = \omega^*_{\text{true}} + \mu v_r v_r^T$$

Has only four DOF because $mu$, in six homogeneous eqn.

Can be removed:

-- more than one rotation, $m=3$ at least;

-- internal parameters included in constrained optimization.

Ambiguities, example rotation are around then around $X, Y$-axis $\alpha_{x,y}$ undetermined, and $Z$-axis, only ratio given. Intrinsic to any motion not a particular auto-calibration algorithm.
Rotating camera calibration

corresponding image points then they are related by $\mathbf{x}^i = \mathbf{H}^i \mathbf{x}$, where $\mathbf{H}^i = K^i R^i (K)^{-1} = \mathbf{H}^i_{\infty}$

**Objective**

Given $m \geq 2$ views acquired by a camera rotating about its centre with fixed or varying internal parameters, compute the parameters of each camera. It is assumed that the rotations are not all about the same axis.

**Algorithm**

(i) **Inter-image homographies:** Compute the homography $\mathbf{H}^i$ between each view $i$ and a reference view such that $\mathbf{x}^i = \mathbf{H}^i \mathbf{x}$ using, for example, algorithm 4.6(p123). Normalize the matrices such that $\det \mathbf{H}^i = 1$.

(ii) **Compute $\omega$:**

- In the case of constant calibration: rewrite the equations $\omega = (\mathbf{H}^i)^{-T} \omega (\mathbf{H}^i)^{-1}$, $i = 1, \ldots, m$ as $\mathbf{Ac} = \mathbf{0}$ where $\mathbf{A}$ is a $6m \times 6$ matrix, and $\mathbf{c}$ the elements of the conic $\omega$ arranged as a 6-vector, or

- For variable calibration parameters, use the equation $\omega^i = (\mathbf{H}^i)^{-T} \omega (\mathbf{H}^i)^{-1}$ to express linear constraints on entries of $\omega^i$ in table 19.4 (e.g. unit aspect ratio) as linear equations in the entries of $\omega$.

(iii) **Compute $K$:** Determine the Cholesky decomposition of $\omega$ as $\omega = \mathbf{UU}^T$, and thence $K = \mathbf{U}^{-T}$.

(iv) **Iterative improvement:** (Optional) Refine the linear estimate of $K$ by minimizing

$$\sum_{i=2, m; j=1, n} d(\mathbf{x}^i_j, K^i R^i K^{-1} \mathbf{x}^j)$$

over $K$ and $R^i$, where $\mathbf{x}^i_j, \mathbf{x}^j$ are the position of the $j$-th point measured in the first and $i$-th images respectively. Initial estimates for the minimization are obtained from $K$ and $R^i = K^{-1} \mathbf{H}^i K$. 
Internal parameters are constant but unknown.

\( K_{\text{linear}} \) and \( K_{\text{iterative}} \) are very close.
Planar homography mosaicing

Panning around the center not zoomed, but auto-focus may change.
Fig. 19.4. Rotation for varying internal parameters assuming square pixels. These are for the panned sequence of figure 8.9(p206). (a) Focal length. (b) Principal point. (c) Pan angle. (d) Tilt angle. Figures courtesy of Lourdes de Agapito Vicente.

Constraints: square pixels -- zero skew, unity square ratio. Gives two linear constraints on omega. Solved with the rotating camera algorithm.
Auto-calibration from planes

It is not possible to determine the depth of the cameras with the usual method. Triggs(1998) gave a solution.

Set of image-to-image homographies, related to first image $H^i$, and can be found easily. The two *mapped* circular points (absolute conic & plane) from the *first view* can be transferred. The calibration matrix $K$ has two eqn. per view in circular points. In each view are two constraints on $\omega^i$. $H^1 = I$

The first view circular points conic equations

Circular points are complex, 4 parameters + $v$ parameters in $K^i$ in $m$ views. To solve it, two per view

$$2m \geq v + 4$$
Many cases the problem is non-linear, and even an iterative solution is difficult. If you know the vanishing line of a plane only two parameters needed to find the *imaged circular points on the line*. Imaged circular points, $\mathbf{K}$ can be determined if an other orthogonal vanishing points is also known. Than, zero skew and square pixels $\implies$ one view.

<table>
<thead>
<tr>
<th>Condition</th>
<th>dof($v$)</th>
<th>views</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown but constant internal parameters</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Constant known skew and aspect ratio. Constant unknown principal point and focal length</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>All internal parameters known except varying focal length</td>
<td>$m$</td>
<td>4</td>
</tr>
<tr>
<td>Varying focal length, all other internal parameters fixed but unknown</td>
<td>$m + 4$</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 19.5. The number of views required for calibration from a plane under various conditions. Calibration is (in principle) possible if $2m \geq v + 4$. 
Auto-calibration of a stereo rig

Relative orientation unchanged during motion. Projective structure $X$ before, $X'$ after, $4x4$ $H_P$

Conjugacy relation: $X_E$ and $X'_E$ are the Euclidean coordinate.

$$X_E = H_{EP} X \quad X'_E = H_{EP} X'$$

$H_{EP}$ is from projective to Euclidean.

$$H_P = H_{EP}^{-1} H_E H_{EP}$$

$H_P$ conjugate to a Euclidean transformation. Same eigenvalues as $H_E$. If $E$ (should be $e$) is eigenvector of $H_E$, satisfies

$$H_{EP} H_P H_{EP}^{-1} E = \lambda E$$

Euclidean transformation: rotation about Z-axis, unit translation along Z-axis. 3D screw motion.

$$H_E = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
\{e^{i\theta}, e^{-i\theta}, 1, 1\} and the corresponding eigenvectors of \(H_E\) are

\[
E_1 = \begin{pmatrix} 1 \\ i \\ 0 \\ 0 \end{pmatrix}, \quad E_2 = \begin{pmatrix} 1 \\ -i \\ 0 \\ 0 \end{pmatrix}, \quad E_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad E_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}
\]

All the eigenvectors lie on \(\pi_\infty\).

Two are the circular points of the planes perpendicular to the rotation (Z) axis. Two are identical, direction of rotation axis. A point transformation \(H_E\) maps to \(H_P\), same eigenvectors.

Eigenvectors of \(H_E^{-T}\) are mapped to those of \(H_P^{-T}\), in the same manner

\(\pi_\infty\) may be computed uniquely as the real eigenvector of \(H_P^{-T}\), or equivalently, and more simply, as the real eigenvector of \(H_P^T\).

---

The two identical eigenvalues geometrically give only one multiplicity since we are in 3D and the motion is non-planar.

---

One motion is enough, but a one-parameters family remains. Only additional motion can eliminate it.
**Objective**

Given two (or more) stereo pairs of images acquired by a fixed stereo rig undergoing general motions (i.e. both R and t are non-zero, and t not perpendicular to the axis of R), compute an affine reconstruction.

**Algorithm**

(i) **Compute an initial projective reconstruction \( X \):** Using the first stereo pair compute a projective reconstruction \((P^L, P^R, \{X_j\})\) as described in chapter 10. This involves computing the fundamental matrix \( F \) and point correspondences between the images of the first pair \( x^L_j \leftrightarrow x^R_j \), e.g. use algorithm 11.4(p291).

(ii) **Compute a projective reconstruction \( X' \) after the motion:** Compute correspondences between the images of the second stereo pair \( x'^L_j \leftrightarrow x'^R_j \). Since both the internal and relative external parameters of the cameras are fixed, the second stereo pair has the same fundamental matrix \( F \) as the first. The same camera matrices \( P^L, P^R \) are used for triangulating points \( X'_j \) in 3-space from the computed correspondences \( x'^L_j \leftrightarrow x'^R_j \) in the second stereo pair.

(iii) **Compute the \( 4 \times 4 \) matrix \( H_p \) which relates \( X \) to \( X' \):** Compute correspondences between the left images of the two stereo pairs \( x^L_j \leftrightarrow x'^L_j \) (e.g. again use algorithm 11.4(p291)). This establishes correspondences between the space points \( X_j \leftrightarrow X'_j \). The homography \( H_p \) may be estimated linearly from five or more of these 3-space point correspondences, and then the estimate refined by minimizing a suitable cost function over \( H_p \). For example, minimizing \( \sum_j (d(x^L_j, P^L H X'_j)^2 + d(x'^R_j, P^R H X'_j)^2) \) minimizes the distance between the measured and reprojected image points.

(iv) **Affine reconstruction:** Compute \( \pi_\infty \) from the real eigenvector of \( H_p^T \) and thence an affine reconstruction.
Fig. 19.9. **Auto-calibration of a stereo rig.** The input stereo pairs before (a) and after (b) the motion of the rig. The stereo rig moves left by about 20 cm, pans by about $10^\circ$ and changes elevation by about $10^\circ$. The accuracy of the computed $H_\infty$ is assessed on another stereo pair acquired by the same rig as follows: In (c), the **left image** (of a stereo pair), a vanishing point is computed by intersecting the imaged sides of the table (which are parallel in the scene). In (d), the **right image** (of a stereo pair), the corresponding vanishing point is computed. The **white square** (near the line intersection) is the vanishing point from the left image mapped to the right using the computed $H_\infty$. In the absence of error the points should be identical. (e) Following **metric calibration** the computed angle between the desk sides (shown in white) from the 3D reconstruction is $90.7^\circ$, in very good agreement with the veridical value.
Auto-calibration is *not* a fool proof method. Works in right circumstances, fails in others.

Recommendations:

-- small motions, restricted motions don't estimate well an infinite homography.

-- use maximum additional information. Zero skew alone is not enough. Even if it is not 100% correct, include equation(s), like the principal points, having low weights.

-- Bundle adjustment (at the end) should have the internal para. bounded between limits. e.g. aspect ratio 0.5 to 3. Added as additional equations, maybe with small weights.

-- General motion less reliable than restricted motions. Example: rotation more reliable than rotation+translation.
Will not cover:

-- duality between points and cameras;
reduced camera coordinated (in camera calibration)
exmp. fundamental matrix 3 pairs and determinant zero.

-- cheirality assuring the images are in the front of the cameras;
Sets boundary conditions on the limits.

-- degenerate configurations;
Geometric constraints of different curves for single or multiple views. Give also the resulting ambiguities.