6 Autocalibration

The aim of *autocalibration* is to compute the internal parameters, starting from weakly calibrated cameras.

More in general, the task is to recover metric properties of camera and/or scene, i.e., to compute a Euclidean reconstruction.

There are two classes of methods:

1. Direct: solve directly for the internal parameters.
2. Stratified: first obtain a projective reconstruction and then transform it to a Euclidean reconstruction (in some cases an affine reconstruction is obtained in between).

The reader is referred to [8] for a review of autocalibration, and to [33, 46, 21, 36, 35, 11] for classical and recent work on the subject.

6.1 Counting argument

Consider m cameras. The difference between the d.o.f. of the multifocal geometry (e.g. 7 for two views) and the d.o.f. of the rigid displacements (e.g. 5 for two views) is the number of independent constraints available for the computation of the intrinsic parameters (e.g. 2 for two views).

The multifocal geometry of m cameras (represented by the m-focal tensor) has $11m - 5$ d.o.f. Proof: a set of m cameras have $11m$ d.o.f., but they determine the m-focal geometry up to a collineation of $\mathbb{P}_3$, which has 15 d.o.f. The net sum is $11m - 5$ d.o.f.

On the other hand, the rigid displacements in m views are described by $6m - 7$ parameters: $3(m-1)$ for rotations, $2(m-1)$ for translations, and $m-2$ ratios of translation norms.

Thus, m *weakly calibrated views give* $5m - 8$ *constraints available for computing the intrinsic parameters*.

Let us suppose that m_k parameters are known and m_c parameters are constant.

The first view introduces $5 - m_k$ unknowns. Every view but the first introduces $5 - m_k - m_c$ unknowns.

Therefore, the unknown intrinsic parameters can be computed provided that

$$5m - 8 \geq (m-1)(5 - m_k - m_c) + 5 - m_k. \tag{98}$$

For example, if the intrinsic parameters are constant, three views are sufficient to recover them.

If one parameter (usually the skew) is known and the other parameters are varying, at least eight views are needed.

6.2 A simple direct method

If we consider two views, two independent constraints are available for the computation of the intrinsic parameters from the fundamental matrix.

Indeed, F has 7 d.o.f, whereas E, which encode the rigid displacement, has only 5 d.o.f. There must be two additional constraint that E must satisfy, with respect to F.

In particular, these constraints stem from the equality of two singular values of the essential matrix (Theorem 4.1) which can be decomposed in two independent polynomial equations.

Let F_{ij} be the (known) fundamental matrix relating views i and j, and let K_i and K_j be the respective (unknown) intrinsic parameter matrices.

The idea of [35] is that the matrix

$$E_{ij} = K_i^T F_{ij} K_j, \tag{99}$$

satisfies the constraints of Theorem 4.1 only if the intrinsic parameters are correct.

Hence, the cost function to be minimized is

$$C(K_i,\ i=1\dots n)=\sum_{i=1}^{n}\sum_{j>n}^{n} w_{ij}\frac{{}^1\sigma_{ij}-{}^2\sigma_{ij}}{{}^1\sigma_{ij}+{}^2\sigma_{ij}}, \tag{100}$$

where ${}^1\sigma_{ij} > {}^2\sigma_{ij}$ are the non zero singular values of E_{ij} and w_{ij} are normalized weight factors (linked to the reliability of the fundamental matrix estimate).

The previous counting argument shows that, in the general case of n views, the $n(n-1)/2$ two-view constraints that can be derived are not independent, nevertheless they can be used as they over-determine the solution.

A non-linear least squares solution is obtained with an iterative algorithm (e.g. Gauss-Newton) that uses analytical derivatives of the cost function.

A starting guess is needed, but this cost function is less affected than others by local minima problems. A globally convergent algorithm based on this cost function is described in [9].

6.3 Stratification

We have seen that a projective reconstruction can be computed starting from points correspondences only (weak calibration), without any knowledge of the camera matrices.

Projective reconstruction differs from Euclidean by an unknown projective transformation in the 3-D projective space, which can be seen as a suitable change of basis.

Starting from a projective reconstruction the problem is computing the transformation that "straighten" it, i.e., that upgrades it to an Euclidean reconstruction.

To this purpose the problem is *stratified* [31, 5] into different representations: depending on the amount of information and the constraints available, it can be analyzed at a projective, affine, or Euclidean level.

Let us assume that a projective reconstruction is available, that is a sequence P_i of $m+1$ camera matrices and a set $\mathbf{M}^j$ of $n+1$ 3-D points such that:

$$\mathbf{m}_i^j \simeq P_i\mathbf{M}^j \quad i=0\dots m, \quad j=0\dots n. \tag{101}$$

Without loss of generality, we can assume that camera matrices writes:

$$P_0=[I\mid\mathbf{0}]; \qquad P_i=[A_i\mid\mathbf{e}_i] \quad \text{for } i=1\dots m \tag{102}$$

We are looking for the a 4×4 nonsingular matrix T that upgrades the projective reconstruction to Euclidean:

$$\mathbf{m}_i^j \simeq \underbrace{P_iT}_{P_i^{\mathrm{E}}}\underbrace{T^{-1}\mathbf{M}^j}_{\text{structure}}, \tag{103}$$

$P_i^{\mathrm{E}}=P_iT$ is the Euclidean camera,

We can choose the first Euclidean-calibrated camera to be $P_0^{\mathrm{E}}=K_0[I\mid\mathbf{0}]$, thereby fixing arbitrarily the world reference frame:

$$P_0^{\mathrm{E}}=K_0[I\mid\mathbf{0}] \qquad P_i^{\mathrm{E}}=K_i[R_i\mid\mathbf{t}_i] \quad \text{for } i=1\dots m. \tag{104}$$

With this choice, it is easy to see that $P_0^{\mathrm{E}}=P_0T$ implies

$$T=\begin{bmatrix}K_0 & \mathbf{0}\\ \mathbf{r}^T & s\end{bmatrix} \tag{105}$$

where $\mathbf{r}^T$ is a 3-vector and s is a scale factor, which we will arbitrarily set to 1 (the Euclidean reconstruction is up to a scale factor).

Under this parameterization T is clearly non singular, and it depends on eight parameters.

Substituting (105) in $P_i^{\mathrm{E}}\simeq P_iT$ gives

$$P_i^{\mathrm{E}}=[K_iR_i\mid K_i\mathbf{t}_i], \simeq P_iT=[A_iK_0+\mathbf{e}_i\mathbf{r}^T\mid\mathbf{e}_i] \quad \text{for } i>0 \tag{106}$$

and, considering only the leftmost 3×3 submatrix, gives

$$K_iR_i\simeq A_iK_0+\mathbf{e}_i\mathbf{r}^T=P_i\begin{bmatrix}K_0\\ \mathbf{r}^T\end{bmatrix} \tag{107}$$

Rotation can be eliminated using $RR^T = I$, leaving:

$$K_i K_i^T \simeq P_i \begin{bmatrix} K_0 K_0^T & K_0 \mathbf{r} \\ \mathbf{r}^T K_0^T & \mathbf{r}^T \mathbf{r} \end{bmatrix} P_i^T \tag{108}$$

This is the basic equation for autocalibration, relating the unknowns K_i $(i = 0 \dots m)$ and $\mathbf{r}$ to the available data P_i (obtained from weakly calibrated images).

Note that (117) contains five equations, because the matrices of both members are symmetric, and the homogeneity reduces the number of equations with one.

6.3.1 Geometric interpretation

Under camera matrix P the outline of the quadric Q is the conic C given by:

$$C^* \simeq P Q^* P^T \tag{109}$$

where C^* is the dual conic and Q^* is the dual quadric. An expression with Q and C may be derived, but it is quite complicated. C^* (resp. Q^*) is the adjoint matrix of C (resp. Q). If C is non singular, then $C^* = C^{-1}$.

In the beginning we introduced the absolute conic $\mathbf{\Omega}$, which is invariant under similarity transformation, hence deeply linked with the Euclidean stratum.

In a Euclidean frame, its equation is $x_1^2 + x_2^2 + x_3^2 = 0 = x_4$.

The absolute conic may be regarded as a special quadric (a disk quadric), therefore its dual is a quadric, the *dual absolute quadric*, denoted by $\mathbf{\Omega}^*$. Its representation is:

$$\mathbf{\Omega}^* = \mathrm{diag}(1, 1, 1, 0). \tag{110}$$

As we already know, the *image of the absolute conic* under camera matrix P^{E} is given by $\boldsymbol{\omega} = (KK^T)^{-1}$, that is :

$$\boldsymbol{\omega}^* = (KK^T) \simeq P^{\mathrm{E}} \mathbf{\Omega}^* P^{\mathrm{E}T} \tag{111}$$

This property is independent on the choice of the projective basis. What changes is the representation of the dual absolute quadric, which is mapped to

$$\mathbf{\Omega}^* = T \mathrm{diag}(1, 1, 1, 0) T^T. \tag{112}$$

under the collineation T.

Substituting T from Eq. (105) into the latter gives:

$$\mathbf{\Omega}^* = \begin{bmatrix} K_0 K_0^T & K_0 \mathbf{r} \\ \mathbf{r}^T K_0^T & \mathbf{r}^T \mathbf{r} \end{bmatrix} \tag{113}$$

Recalling that $\boldsymbol{\omega}_i^* = K_i K_i^T$, then Eq. (108) is equivalent to

$$\boldsymbol{\omega}_i^* \simeq P_i \mathbf{\Omega}^* P_i^T \tag{114}$$

6.3.2 Solution strategies

Autocalibration requires to solve Eq. (114), with respect with $\mathbf{\Omega}^*$ (and $\boldsymbol{\omega}_i^*$).

If $\mathbf{\Omega}^*$ is known, the collineation T that upgrades cameras from projective to Euclidean is obtained by decomposing $\mathbf{\Omega}^*$ as in Eq. (112), using the eigenvalue decomposition.

$\mathbf{\Omega}^*$ might be parameterized as in Eq. (113) with 8 d.o.f. or parameterized as a generic 4×4 symmetric matrix (10 d.o.f.). The latter is an over-parameterization, as $\mathbf{\Omega}^*$ is also singular and defined up to a scale factor (which gives again 8 d.o.f.).

There are several strategies for dealing with the scale factor.

- Introduce the scale factor explicitly as an additional unknown [20]:

$$\boldsymbol{\omega}_i^* - \lambda_i P_i \mathbf{\Omega}^* P_i^T = \mathbf{0} \tag{115}$$

  This gives 6 equations but introduces one additional unknown (the net sum is 5).

- Eliminate it by using the same idea of the cross product for 3-vectors [46]:

$$\operatorname{vec}\boldsymbol{\omega}_i^* \simeq \operatorname{vec}(P_i\boldsymbol{\Omega}^*P_i^T) \iff \operatorname{rank}\left[\operatorname{vec}\boldsymbol{\omega}_i^* | \operatorname{vec}(P_i\boldsymbol{\Omega}^*P_i^T)\right] = 1$$

  which is tantamount to say that every 2×2 minor of $[\operatorname{vec}\boldsymbol{\omega}_i^* | \operatorname{vec}(P_i\boldsymbol{\Omega}^*P_i^T)]$ is zero. As matrices in Eq. (114) are symmetric, only 6 elements need to be considered in the corresponding vectors. There are 15 different order-2 minors of a 6×2 matrix, but only 5 equations are independent.

- Use a matrix norm (namely, Frobenius norm) [36]:

$$\frac{\boldsymbol{\omega}_i^*}{||\boldsymbol{\omega}_i^*||_F} - \frac{P_i\boldsymbol{\Omega}^*P_i^T}{||P_i\boldsymbol{\Omega}^*P_i^T||_F} = \mathbf{0} \tag{116}$$

In any case, a non-linear least-squares problem has to be solved. Available numerical techniques (based on the Gauss-Newton method) are iterative, and requires an estimate of the solution to start.

This can be obtained by doing an educated guess about skew, principal point and aspect ratio, and solve the linear problem that results.

Linear solution

If some of the internal parameters are known, this might cause some elements of $\boldsymbol{\omega}_i^*$ to vanish. Linear equations on $\boldsymbol{\Omega}^*$ are generated from zero-entries of $\boldsymbol{\omega}^*$ (because this eliminates the scale factor):

$$\boldsymbol{\omega}_i^*(k,\ell) = 0 \;\Rightarrow\; \mathbf{p}_{i,k}^T\boldsymbol{\Omega}^*\mathbf{p}_{i,\ell} = 0$$

where $\mathbf{p}_{i,k}^T$ is the k-th row of P_i.

Likewise, linear constraints on $\boldsymbol{\Omega}^*$ can be obtained from the equality of elements in the the upper (lower) triangular part of $\boldsymbol{\omega}_i^*$ (because $\boldsymbol{\omega}_i^*$ is symmetric).

In order to be able to solve linearly for $\boldsymbol{\Omega}^*$, at least 10 linear equations must be stacked up, to form a homogeneous linear system, which can be solved as usual (via SVD). Singularity of $\boldsymbol{\Omega}^*$ can be enforced a-posteriori by forcing the smallest singular value to zero.

If the principal point is known, $\boldsymbol{\omega}_i^*(1,3) = 0 = \boldsymbol{\omega}_i^*(2,3)$ and this gives two linear constraints. If, in addition, skew is zero we have $\boldsymbol{\omega}_i^*(1,2) = 0$. Known aspect ratio r provides a further constraint: $r\boldsymbol{\omega}_i^*(1,1) = \boldsymbol{\omega}_i^*(2,2)$.

Constant internal parameters

If all the cameras has the same internal parameters, so $K_i = K$, then Eq. (108) becomes

$$KK^T \simeq P_i \begin{bmatrix} KK^T & K\mathbf{r} \\ \mathbf{r}^TK^T & \mathbf{r}^T\mathbf{r} \end{bmatrix} P_i^T \tag{117}$$

The constraints expressed by Eq. (117) are called the Kruppa constraints in [20].

Since each camera matrix, apart from the first one, gives five equations in the eight unknowns, a unique solution is obtained when at least three views are available.

The resulting system of equations is solved with a non-linear least-squares technique (e.g. Gauss-Newton).

7 Getting practical

In this section we will approach estimation problems from a more "practical" point of view.

First, we will discuss how the presence of errors in the data affects our estimates and describe the countermeasures that must be taken to obtain a good estimate.

Finally, we introduce non-linear distortions due to lenses into the pinhole model and we illustrate a practical calibration algorithm that works with a simple planar object.

7.1 Pre-conditioning

In presence of noise (or errors) on input data, the accuracy of the solution of a linear system depends crucially on the *condition number* of the system. The lower the condition number, the less the input error gets amplified (the system is more stable).

As [14] pointed out, it is crucial for linear algorithms (as the DLT algorithm) that input data is properly pre-conditioned, by a suitable coordinate change (origin and scale): points are translated so that their centroid is at the origin and are scaled so that their average distance from the origin is $\sqrt{2}$.

This improves the condition number of the linear system that is being solved.

Apart from improved accuracy, this procedure also provides invariance under similarity transformations in the image plane.

7.2 Algebraic vs geometric error

Measured data (i.e., image or world point positions) is noisy.

Usually, to counteract the effect of noise, we use more equations than necessary and solve with least-squares.

What is actually being minimized by least squares?

In a typical null-space problem formulation $Ax = 0$ (like the DLT algorithm) the quantity that is being minimized is the square of the residual $||Ax||$.

In general, if $||Ax||$ can be regarded as a distance between the geometrical entities involved (points, lines, planes, etc..), than what is being minimized is a geometric error, otherwise (when the error lacks a good geometrical interpretation) it is called an algebraic error.

All the linear algorithm (DLT and others) we have seen so far minimize an algebraic error. Actually, there is no justification in minimizing an algebraic error apart from the ease of implementation, as it results in a linear problem.

Usually, the minimization of a geometric error is a non-linear problem, that admit only iterative solutions and requires a starting point.

So, why should we prefer to minimize a geometric error? Because:

- The quantity being minimized has a meaning
- The solution is more stable
- The solution is invariant under Euclidean transforms

Often linear solution based on algebraic residuals are used as a starting point for a non-linear minimization of a geometric cost function, which gives the solution a final "polish" [12].

7.2.1 Geometric error for resection

The goal is to estimate the camera matrix, given a number of correspondences $(\mathbf{m}^j, \mathbf{M}^j) \quad j = 1 \dots n$

The geometric error associated to a camera estimate $\hat{P}$ is the distance between the measured image point $\mathbf{m}^j$ and the re-projected point $\hat{P}_i \mathbf{M}^j$:

$$\min_{\hat{P}} \sum_j d(\hat{P}\mathbf{M}^j, \mathbf{m}^j)^2 \tag{118}$$

where $d()$ is the Euclidean distance between the homogeneous points.

The DLT solution is used as a starting point for the iterative minimization (e.g. Gauss-Newton)

7.2.2 Geometric error for triangulation

The goal is to estimate the 3D coordinates of a point $\mathbf{M}$, given its projection $\mathbf{m}_i$ and the camera matrix $\mathbf{P}_i$ for every view $i = 1 \dots m$.

The geometric error associated to a point estimate $\hat{\mathbf{M}}$ in the i-th view is the distance between the measured image point $\mathbf{m}_i$ and the re-projected point $P_i\hat{\mathbf{M}}$:

$$\min_{\hat{\mathbf{M}}} \sum_i d(P_i\hat{\mathbf{M}}, \mathbf{m}_i)^2 \qquad (119)$$

where $d()$ is the Euclidean distance between the homogeneous points.

The DLT solution is used as a starting point for the iterative minimization (e.g. Gauss-Newton).

7.2.3 Geometric error for F

The goal is to estimate F given a a number of point correspondences $\mathbf{m}_\ell^i \leftrightarrow \mathbf{m}_r^i$.

The geometric error associated to an estimate $\hat{F}$ is given by the distance of conjugate points from conjugate lines (note the symmetry):

$$\min_{\hat{F}} \sum_j d(\hat{F}\mathbf{m}_\ell^j, \mathbf{m}_r^j)^2 + d(\hat{F}^T\mathbf{m}_r^j, \mathbf{m}_\ell^j)^2 \qquad (120)$$

where $d()$ here is the Euclidean distance between a line and a point (in homogeneous coordinates).

The eight-point solution is used as a starting point for the iterative minimization (e.g. Gauss-Newton).

Note that F must be suitably parameterized, as it has only seven d.o.f.

7.2.4 Geometric error for H

The goal is to estimate H given a a number of point correspondences $\mathbf{m}_\ell^i \leftrightarrow \mathbf{m}_r^i$.

The geometric error associated to an estimate $\hat{H}$ is given by the symmetric distance between a point and its transformed conjugate:

$$\min_{\hat{H}} \sum_j d(\hat{H}\mathbf{m}_\ell^j, \mathbf{m}_r^j)^2 + d(\hat{H}^{-1}\mathbf{m}_r^j, \mathbf{m}_\ell^j)^2 \qquad (121)$$

where $d()$ is the Euclidean distance between the homogeneous points. This also called the *symmetric transfer error*.

The DLT solution is used as a starting point for the iterative minimization (e.g. Gauss-Newton).

7.2.5 Bundle adjustment (projective reconstruction)

If measurements are noisy, the projection equation will not be satisfied exactly by the camera matrices and structure computed in Sec. 5.3.2.

We wish to minimize the image distance between the re-projected point $\hat{P}_i\hat{\mathbf{M}}^j$ and measured image points $\mathbf{m}_i^j$ for every view in which the 3D point appears:

$$\min_{\hat{P}_i, \hat{\mathbf{M}}^j} \sum_{i,j} d(\hat{P}_i\hat{\mathbf{M}}^j, \mathbf{m}_i^j)^2 \qquad (122)$$

where $d()$ is the Euclidean distance between the homogeneous points.

As m and n increase, this becomes a very large minimization problem.

A solution is to alternate minimizing the re-projection error by varying $\hat{P}_i$ with minimizing the re-projection error by varying $\hat{\mathbf{M}}^j$.

7.2.6 Bundle adjustment (autocalibration)

If a Euclidean reconstruction ha been obtained from autocalibration, bundle adjustment can be applied to refine structure and calibration (i.e., Euclidean camera matrices):

$$\min_{\hat{P}_i, \hat{\mathbf{M}}^j} \sum_{i,j} d(\hat{P}_i \hat{\mathbf{M}}^j, \mathbf{m}_i^j)^2 \tag{123}$$

where $\hat{P}_i = \hat{K}_i[\hat{R}_i|\hat{\mathbf{t}}_i]$ and the rotation has to be suitably parameterized (e.g. quaternions) parameterized with three parameters (see Rodrigues formula).

See [47] for a review and a more detailed discussion on bundle adjustment.

7.3 Robust estimation

Up to this point, we have assumed that the only source of error affecting correspondences is in the measurements of point's position. This is a small-scale noise that gets averaged out with least-squares.

In practice, we can be presented with *mismatched* points, which are *outliers* to the noise distribution (i.e., wrong measurements following a different, unmodelled, distribution).

These outliers can severely disturb least-squares estimation (even a single outlier can totally offset the least-squares estimation, as demonstrated in Fig. 16.)

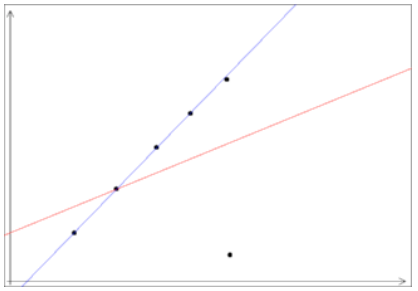


Fig. 16. A single outlier can severely offset the least-squares estimate (red line), whereas the robust estimate (blue line) is unaffected.

The goal of robust estimation is to be insensitive to outliers (or at least to reduce sensitivity).

7.3.1 M-estimators

Least squares:

$$\min_{\theta} \sum_i (r_i/\sigma_i)^2 \tag{124}$$

where θ are the regression coefficient (what is being estimated) and r_i is the residual. M-estimators are based on the idea of replacing the squared residuals by another function of the residual, yielding

$$\min_{\theta} \sum_i \rho(r_i/\sigma_i) \tag{125}$$

ρ is a symmetric function with a unique minimum at zero that grows sub-quadratically, called *loss function*.

Differentiating with respect to θ yields:

$$\sum_i \frac{1}{\sigma_i} \rho'(r_i/\sigma_i) \frac{dr_i}{d\theta} = 0 \tag{126}$$

The M-estimate is obtained by solving this system of non-linear equations.

7.3.2 RANSAC

Given a model that requires a minimum of p data points to instantiate its free parameters θ, and a set of data points S containing outliers:

1. Randomly select a subset of p points of S and instantiate the model from this subset
2. Determine the set S_i of data points that are within an error tolerance t of the model. S_i is the consensus set of the sample.
3. If the size of S_i is greater than a threshold T, re-estimate the model (possibly using least-squares) using S_i (the set of inliers) and terminate.
4. If the size of S_i is less than T, repeat from step 1.
5. Terminate after N trials and choose the largest consensus set found so far.

Three parameters need to be specified: t, T and N.

Both T and N are linked to the (unknown) fraction of outliers ϵ.

N should be large enough to have a high probability of selecting at least one sample containing all inliers. The probability to randomly select p inliers in N trials is:

$$P = 1 - (1 - (1 - \epsilon)^p)^N \tag{127}$$

By requiring that P must be near 1, N can be solved for given values of p and ϵ.

T should be equal to the expected number of inliers, which is given (in fraction) by $(1 - \epsilon)$.

At each iteration, the largest consensus set found so fare gives a lower bound on the fraction of inliers, or, equivalently, an upper bound on the number of outliers. This can be used to adaptively adjust the number of trials N.

t is determined empirically, but in some cases it can be related to the probability that a point under the threshold is actually an inlier [12].

As pointed out in [41], RANSAC can be viewed as a particular M-estimator.

The objective function that RANSAC maximizes is the number of data points having absolute residuals smaller that a predefined value t. This may be seen a minimising a binary loss function that is zero for small (absolute) residuals, and 1 for large absolute residuals, with a discontinuity at t.

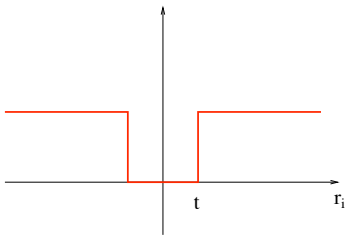


Fig. 17. RANSAC loss function

By virtue of the prespecified inlier band, RANSAC can fit a model to data corrupted by substantially more than half outliers.

7.3.3 LMedS

Another popular robust estimator is the Least Meadian of Squares. It is defined by:

$$\min_{\theta} \text{med}_i r_i \tag{128}$$

It can tolerate up to 50% of outliers, as up to half of the data point can be arbitrarily far from the "true" estimate without changing the objective function value.

Since the median is not differentiable, a random sampling strategy similar to RANSAC is adopted. Instead of using the consensus, each sample of size p is scored by the median of the residuals of all the data points. The model with the least median (lowest score) is chosen.

A final weighted least-squares fitting is used.

With respect to RANSAC, LMedS can tolerate "only" 50% of outliers, but requires no setting of thresholds.

7.4 Practical calibration

Camera calibration (or resection) as described so far, requires a calibration object that consists typically of two or three planes orthogonal to each other. This might be difficult to obtain, without access to a machine tool.

Zhang [50] introduced a calibration technique that requires the camera to observe a planar pattern (much easier to obtain) at a few (at least three) different orientation. Either the camera or the planar pattern can be moved by hand.

Instead of requiring one image of many planes, this method requires many images of one plane.

We will also introduce here a more realistic camera model that takes into account non-linear effects produced by lenses.

In each view, we assume that correspondences between image points and 3D points on the planar pattern have been established.

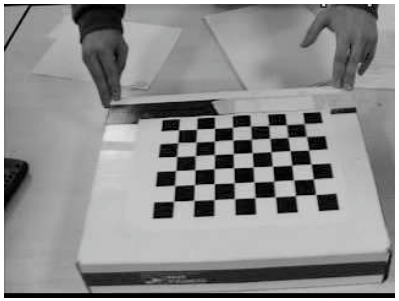


Fig. 18. Image of a planar calibration pattern. The points used for calibration are the corners of the black squares.

7.4.1 Estimating internal parameters

Following the development of Sec. 4.4 we know that for a camera $P = K[R|\mathbf{t}]$ the homography between a world plane at $z = 0$ and the image is

$$H \simeq K[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}] \tag{129}$$

where $\mathbf{r}_i$ are the column of R.

Suppose that H is computed from correspondences bwtween four or more known world points and their images, then some constraints can be obtained on the intrinsic parameters, thanks to the fact that the columns of R are orthonormal.

Writing $H = [\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3]$, from the previous equation we derive:

$$\mathbf{r}_1 = \lambda K^{-1}\mathbf{h}_1 \quad \text{and} \quad \mathbf{r}_2 = \lambda K^{-1}\mathbf{h}_2 \tag{130}$$

where λ is an unknown scale factor.

The orthogonality $\mathbf{r}_1^T\mathbf{r}_2 = 0$ gives

$$\lambda^2\mathbf{h}_1^T(KK^T)^{-1}\mathbf{h}_2 = 0 \tag{131}$$

or, equivalently (remember that $\boldsymbol{\omega} = (KK^T)^{-1}$)

$$\mathbf{h}_1^T\boldsymbol{\omega}\mathbf{h}_2 = 0 \tag{132}$$

Likewise, the condition on the norm $\mathbf{r}_1^T\mathbf{r}_1 = \mathbf{r}_2^T\mathbf{r}_2$ gives

$$\mathbf{h}_1^T\boldsymbol{\omega}\mathbf{h}_1 = \mathbf{h}_2^T\boldsymbol{\omega}\mathbf{h}_2 \tag{133}$$

Introducing the Kronecker product as usual, we rewrite these two equations as:

$$(\mathbf{h}_1^T \otimes \mathbf{h}_2^T)\,\mathrm{vec}\,\boldsymbol{\omega} = 0 \tag{134}$$

$$\big((\mathbf{h}_1^T \otimes \mathbf{h}_1^T) - (\mathbf{h}_2^T \otimes \mathbf{h}_2^T)\big)\,\mathrm{vec}\,\boldsymbol{\omega} = 0 \tag{135}$$

A single view of the plane gives two equations in six unknowns, hence a solution is achievable with $n \geq 3$ views (in practice, for a good calibration, one should use around 12 views).

K is obtained from the Cholesky factorization of $\boldsymbol{\omega}$, then R and $\mathbf{t}$ are recovered from:

$$[\mathbf{r}_1, \mathbf{r}_2, \mathbf{t}] = \frac{1}{||K^{-1}\mathbf{h}_1||}K^{-1}[\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3] \qquad \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \tag{136}$$

Because of noise, the matrix R is not guaranteed to be orthogonal, hence we need to recover the closest orthogonal matrix.

Let $R = QS$ be the polar decomposition of R. Then Q is the closest possible orthogonal matrix to R in Frobenius norm.

In this way we have obtained the camera matrix P by minimizing an algebraic distance which is not geometrically meaningful.

It is advisable to refine it with a (non-linear) minimization of a geometric error:

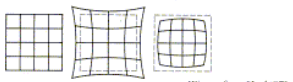
$$\min_{\hat{P}_i} \sum_{i=1}^{n} \sum_{j=1}^{m} d(\hat{P}_i \mathbf{M}^j, \mathbf{m}_i^j)^2 \tag{137}$$

where $\hat{P}_i = \hat{K}[\hat{R}_i | \hat{\mathbf{t}}_i]$ and the rotation has to be suitably parameterized with three parameters (see Rodrigues formula).

The linear solution is used as a starting point for the iterative minimization (e.g. Gauss-Newton).

7.4.2 Radial distortion

A realistic model for a photocamera or a videocamera must take into account non-linear distortions introduced by the lenses, especially when dealing with short focal lengths or low cost devices (e.g. webcams, disposable cameras).



[Figure from Hecht87]

The more relevant effect is the *radial distortion*, which is modeled as a non-linear transformation from ideal (undistorted) coordinates (u, v) to real observable (distorted) coordinates $(\hat{u}, \hat{v})$:

$$\begin{cases} \hat{u} = (u - u_0)(1 + k_1 r_d^2) + u_0 \\ \hat{v} = (v - v_0)(1 + k_1 r_d^2) + v_0 \end{cases} . \tag{138}$$

where $r_d^2 = \left(\frac{(u-u_0)}{a_u}\right)^2 + \left(\frac{(v-v_0)}{a_v}\right)^2$ and (u_0, v_0) are the coordinates of the image centre.

Estimating k_1

Let us assume that the pinhole model is calibrated. The point $\mathbf{m} = (u, v)$ projected according to the pinhole model (undistorted) do not coincide with the measured points $\hat{\mathbf{m}} = (\hat{u}, \hat{v})$ because of the radial distortion.

We wish to recover k_1 from Eq. (138). Each point gives two equation:

$$\begin{cases} (u - u_0)\left(\left(\frac{(u-u_0)}{a_u}\right)^2 + \left(\frac{(v-v_0)}{a_v}\right)^2\right) k_1 = \hat{u} - u \\ (v - u_0)\left(\left(\frac{(u-u_0)}{a_u}\right)^2 + \left(\frac{(v-v_0)}{a_v}\right)^2\right) k_1 = \hat{v} - v \end{cases} \tag{139}$$

hence a least squares solution for k_1 is readily obtained from $n > 1$ points.

When calibrating a camera we are required to *simultaneously* estimate both the pinhole model's parameters and the radial distortion coefficient.

The pinhole calibration we have described so far assumed no radial distortion, and the radial distortion calibration assumed a calibrated pinhole camera.

The solution (a very common one in similar cases) is to alternate between the two estimation until convergence.

Namely: start assuming $k = 0$, calibrate the pinhole model, then use that model to compute radial distortion. Once k_1 is estimated, refine the pinhole model by solving Eq. (137) with the radial distorion in the projection, and continue until the image error is small enough.

8 Further readings

General books on (Geometric) Computer Vision are: [4, 48, 6, 12].

Acknowledgements

The backbone of this tutorial has been adapted from [25]. Many ideas are taken from [12]. Most of the material comes from multiple sources, but some sections are based on one or two works. In particular: the section on rectification is adapted from [10]; Sec. 5.2 is based on [39]: Sec. 5.4 is based on [19].

Michela Farenzena, Spela Ivekovic, Alessandro Negrente, Sara Ceglie and Roberto Marzotto produced most of the pictures. Some of them were inspired by [51].

References

[1] S. Avidan and A. Shashua. Novel view synthesis in tensor space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1034–1040, 1997.

[2] P. Beardsley, A. Zisserman, and D. Murray. Sequential update of projective and affine structure from motion. *International Journal of Computer Vision*, 23(3):235–259, 1997.

[3] B. S. Boufama. The use of homographies for view synthesis. In *Proceedings of the International Conference on Pattern Recognition*, pages 563–566, 2000.

[4] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, MA, 1993.

[5] O. Faugeras. Stratification of 3-D vision: projective, affine, and metric representations. *Journal of the Optical Society of America A*, 12(3):465–484, 1994.

[6] O. Faugeras and Q-T Luong. *The geometry of multiple images*. MIT Press, 2001.

[7] O. D. Faugeras and L. Robert. What can two images tell us about a third one? In *Proceedings of the European Conference on Computer Vision*, pages 485–492, Stockholm, 1994.

[8] A. Fusiello. Uncalibrated Euclidean reconstruction: A review. *Image and Vision Computing*, 18(6-7):555–563, May 2000.

[9] A. Fusiello, A. Benedetti, M. Farenzena, and A. Busti. Globally convergent autocalibration using interval analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1633–1638, December 2004.

[10] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.

[11] R. Hartley, E. Hayman, L. de Agapito, and I. Reid. Camera calibration and the search for infinity. In *Proceedings of the IEEE International Conference on Computer Vision*, 1999.

[12] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2nd edition, 2003.

[13] R. I. Hartley. Estimation of relative camera position for uncalibrated cameras. In *Proceedings of the European Conference on Computer Vision*, pages 579–587, Santa Margherita L., 1992.

[14] R. I. Hartley. In defence of the 8-point algorithm. In *Proceedings of the IEEE International Conference on Computer Vision*, 1995.

[15] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, November 1997.

[16] R.I. Hartley. Theory and practice of projective rectification. *International Journal of Computer Vision*, 35(2):1–16, November 1999.

[17] A. Heyden. Multiple view geometry using multifocal tensors.

[18] A. Heyden. Projective structure and motion from image sequences using subspace methods. In *Scandinavian Conference on Image Analysis*, 1997.

[19] A. Heyden. Tutorial on multiple view geometry. In conjunction with ICPR00, September 2000.

[20] A. Heyden and K. Åström. Euclidean reconstruction from constant intrinsic parameters. In *Proceedings of the International Conference on Pattern Recognition*, pages 339–343, Vienna, 1996.

[21] A. Heyden and K. Åström. Minimal conditions on intrinsic parameters for Euclidean reconstruction. In *Proceedings of the Asian Conference on Computer Vision*, Hong Kong, 1998.

[22] T.S. Huang and O.D. Faugeras. Some properties of the E matrix in two-view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12):1310–1312, December 1989.

[23] F. Isgro and E. Trucco. Projective rectification without epipolar geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages I:94–99, Fort Collins, CO, June 23-25 1999.

[24] F. Isgrò, E. Trucco, P. Kauff, and O. Schreer. 3-D image processing in the future of immersive media. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(3):288–303, 2004.

[25] S. Ivekovic, A. Fusiello, and E. Trucco. Fundamentals of multiple view geometry. In O. Schreer and T. Sikora P. Kauff, editors, *3D Videocommunication. Algorithms, concepts and real-time systems in human centered communication*, chapter 6. John Wiley & Sons, 2005. In press.

[26] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford University Press, 1993.

[27] S. Laveau and O. Faugeras. 3-D scene representation as a collection of images and foundamental matrices. Technical Report 2205, INRIA, Institut National de Recherche en Informatique et an Automatique, February 1994.

[28] Jed Lengyel. The convergence of graphics and vision. *IEEE Computer*, 31(7):46–53, July 1998.

[29] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–488, 1998.

[30] C. Loop and Z. Zhang. Computing rectifying homographies for stero vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages I:125–131, Fort Collins, CO, June 23-25 1999.

[31] Q.-T. Luong and T. Viéville. Canonical representations for the geometries of multiple projective views. *Computer Vision and Image Understanding*, 64(2):193–229, 1996.

[32] S. Mahamud, M. Hebert, Y. Omori, and J. Ponce. Provably-convergent iterative methods for projective structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages I:1018–1025, 2001.

[33] S. J. Maybank and O. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[34] P.R.S. Mendonça. PhD thesis.

[35] P.R.S. Mendonça and R. Cipolla. A simple techinique for self-calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages I:500–505, 1999.

[36] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 90–95, Bombay, 1998.

[37] L. Robert, C. Zeller, O. Faugeras, and M. Hébert. Applications of non-metric vision to some visually-guided robotics tasks. In Y. Aloimonos, editor, *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, chapter 5, pages 89–134. Lawrence Erlbaum Associates, 1997.

[38] A. Shashua. Trilinear tensor: The fundamental construct of multiple-view geometry and its applications. In *International Workshop on Algebraic Frames For The Perception Action Cycle (AFPAC)*, Kiel Germany, Sep. 8-9 1997.

[39] A. Shashua. Multi-view geometry from a stationary scene. Lecture given at U. of Milano, June 2004.

[40] A. Shashua and N. Navab. Relative affine structure: Canonical model for 3D from 2D geometry and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):873–883, September 1996.

[41] C. V. Stewart. Robust parameter estimaton in computer vision. *SIAM Review*, 41(3):513–537, 1999.

[42] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proceedings of the European Conference on Computer Vision*, pages 709–720, Cambridge, UK, 1996.

[43] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, 16(2):22–30, March 1996.

[44] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography – a factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992.

[45] Philip H. S. Torr and Andrew Zisserman. Robust computation and parametrization of multiple view relations. In *ICCV*, pages 727–732, 1998.

[46] B. Triggs. Autocalibration and the absolute quadric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 609–614, Puerto Rico, 1997.

[47] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms*, pages 298–372. Springer-Verlag, 2000.

[48] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall, 1998.

[49] Cha Zhang and Tsuhan Chen. A survey on image-based rendering - representation, sampling and compression. Technical Report AMP 03-03, Electrical and Computer Engineering - Carnegie Mellon University, Pittsburgh, PA 15213, June 2003.

[50] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, March/April 1998.

[51] A. Zisserman. Single view and two-view geometry. Handout, EPSRC Summer School on Computer Vision, 1998. available from http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/EPSRC_SSAZ/epsrc_ssaz.html.