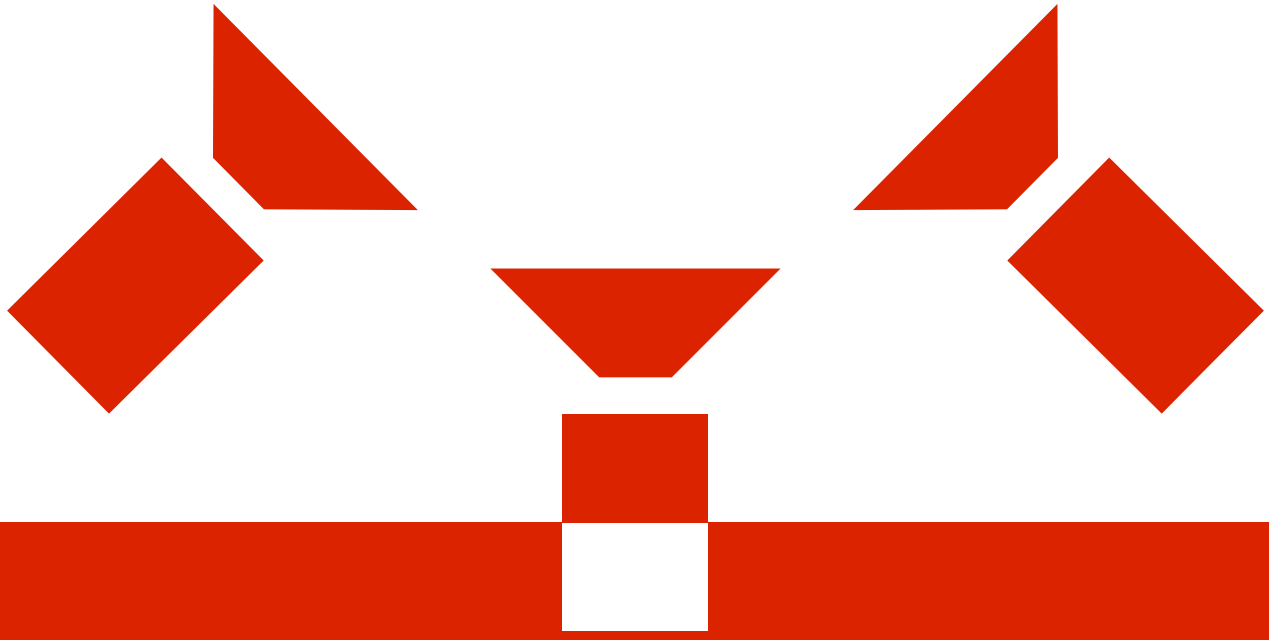
AUTOMATIC MULTIPLE CAMERA CALIBRATION



Project Report of Andreas Geiger
Ecole Polytechnique Fédérale de Lausanne
CVLAB

AUTOMATIC MULTIPLE CAMERA CALIBRATION

USING A PLANAR CALIBRATION PATTERN

Semester Project – Winter 2005/06

by Andreas Geiger
February 17th 2006

Supervisor: Vincent Lepetit, Julien Pilet
Prof. responsible: Pascal Fua, CVLAB

Ecole Polytechnique Fédérale de Lausanne
Computer Vision Laboratory (CVLAB)

I ABSTRACT

Goal of this work was to find and implement a fast and flexible way to perform multiple camera calibration. For a given number of cameras the intrinsic (calibration matrix) and extrinsic (relative rotation and translation) parameters should be extracted, starting from a set of image sequences. The calibration object simply consists of a planar pattern which has to be presented to the cameras at a few (at least two) different orientations and can be produced with every standard laser printer. Therefore this technique is much more easy and flexible compared to classical techniques, which make use of expensive equipment such as two or three orthogonal orientated planes.

For single camera calibration two different techniques coexist up to now: photogrammetric calibration (involving a calibration pattern) and self-calibration (constraints arise from rigidity of the scene while moving the camera around). While the latter is very flexible its drawbacks are results which didn't proof to be reliable at each time. Therefore we concentrate on photogrammetric calibration.

Tsai provides in [9] a photogrammetric “two-stage” calibration technique, which is able to calibrate a camera in real-time even with respect for distortion without using computational intensive non-linear optimization methods. The first stage includes computation of 3D orientation and position. The second stage computes the effective focal length as well as the distortion coefficients. A so called *radial alignment constraint* is used in order to reduce the dimensionality of the unknown parameter space. Nevertheless the drawback of this technique is the requirement of knowing the movement of the calibration pattern or the camera which is not needed by Zhang's technique [3]. Therefore the latter one is more flexible and allows easy estimation of the intrinsic camera's parameters without the use of expensive calibration equipment. A comparison between Tsai's and Zhang's technique (the latter is used in this work for estimating an initial guess) can be found in [10].

Another technique, which can be considered as state-of-the-art, is the direct linear transformation (DLT) [11] developed by Aziz and Karara in 1971. The idea is to observe a point's projection using its homogeneous coordinates:

$$m = PM \quad with \quad m = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}, \quad M = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (1)$$

By normalizing the point on the image plane we can explicitly express (1) by:

$$u = \frac{P_{11}X + P_{12}Y + P_{13}Z + P_{14}}{P_{31}X + P_{32}Y + P_{33}Z + P_{34}} \quad \wedge \quad v = \frac{P_{21}X + P_{22}Y + P_{23}Z + P_{24}}{P_{31}X + P_{32}Y + P_{33}Z + P_{34}} \quad (2)$$

Equations (2) provide us with 2 lines for our equation system we have to solve:

$$\begin{pmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -Xu & -Yu & -Zu & -u \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -Xv & -Yv & -Zv & -v \end{pmatrix} \begin{pmatrix} P_{11} \\ \dots \\ P_{34} \end{pmatrix} = 0 \quad (3)$$

Since we have to determine 12 parameters we need at least 6 of these equation couples which means at least 6 feature points (which do not lie on a plane) in order to calibrate the system. In order to solve (3) we may calculate the eigenvector of the matrix on the left side, corresponding to the minimal eigenvalue. After having determined the projection matrix P it is easy to decompose it into calibration, rotation and translation matrix. Let P_3 be the matrix containing the first 3 columns of P. It follows that

$$P_3 P_3^T = (KR)(KR)^T = KRR^T K^T = KK^T \quad (4)$$

with K being the calibration matrix and R standing for the orthonormal ($R^T = R^{-1}$) rotation matrix. Due to the special upper-triangle form of K we may use Cholesky decomposition to decompose KK^T to K and K^T .

Drawbacks of DLT are on the one side that adding the requirement of distortion parameters transforms the problem in one that has to be solved with non-linear optimization. On the other hand planar calibration objects may not be used except for a situation with an a-priori knowledge of their transformation and rotation.

In our work we focus on calibrating multiple cameras at the same time in order to increase calibration accuracy. Interesting applications in this field are currently tested at Stanford University, for instance. The “Stanford Multi-Camera Array” [12],[13] - consisting of 128 video cameras - is calibrated using similar methods like the ones used here.

The theoretical part of this work can be split into three sections: The detection of feature points in the image followed by calculating the homographies, the guess of intrinsic parameters (which serves as a starting point for optimization) and the estimation of the intrinsic and extrinsic parameters using an overall optimization. For the first part we use

an algorithm proposed by Vincent Lepetit [1],[2]. Estimation of the initial intrinsic camera parameters is done using a filter proposed in this work, combined with a new flexible technique [3], first introduced by Zhengyou Zhang in 1998 and further studied by Peter F. Sturm and Stephen J. Maybank [4]. For the final step, we rearrange the homography mappings and perform a non-linear optimization using the Levenberg-Marquardt algorithm [6]. Afterwards all intrinsic and extrinsic parameters can be extracted easily.

To simplify matters we assume our skew factor to be zero and don't handle any type of distortion. Nevertheless this extension could be added to our system to provide precise tackling of fish eye cameras or cameras exhibiting manufacturer inaccuracies.

II Estimating the Homographies

A homography can be modeled as a 3x3 matrix mapping a 2D homogeneous vector to another 2D homogeneous vector. In order to obtain homographies between the planar calibration pattern (z-coordinate equals zero) and the screen image, we use an algorithm proposed by V. Lepetit [1] using randomized trees for real-time key point recognition.

Randomized trees are used because they are fast enough to do real-time feature point recognition and they are more robust compared to other wide baseline matching methods. Once potential correspondences have been established between the interest points, a standard RANSAC (Random Sample Cosenus Algorithm, performs robust fitting even with large number of outliers) is applied.

The procedure consists of 2 phases: A 'training phase' for finding the points of interest which usually lasts for at least 5 minutes on a standard PC and a 'recognition phase' capable to do real-time recognition using the grown up trees from the training phase. During training a set K of N prominent key points lying on the object is being constructed. At runtime it is possible to quickly decide if an input patch centered at a key point belongs to the object descending the trees. In this method the key points are extracted out of the synthesized views.

For the randomized trees each non-terminal node contains a simple test that splits the image space („Is this pixel brighter than this one?“). A new image is classified by dropping it down the tree, and, in the one tree case, attributing it the class with the maximal conditional probability stored in the leaf it reaches. Due to the fact that the classification relies on tests comparing intensities it is very robust to illumination changes. Tests performed in [1] revealed that 20 trees are enough to reach a recognition rate of 80%. For the node ternary tests based on the difference of intensities of two pixels are used with a threshold τ deciding in which range two intensities should be considered as similar. Once the randomized trees are built, the posterior distributions can be estimated for each terminal node from the training set. Compared to the SIFT approach of David Lowe the algorithm of V. Lepetit shows much better performance when much perspective distorts the object image. It works very well for textured objects but loses its effectiveness in the absence of texture. For our calibration problem this doesn't pose a problem since we use a planar calibration pattern which is a textured image.
Further, the algorithm was slightly modified in the sense that we introduced 2 threshold parameters: The first (Distance_Max) is specifying the maximal error of an inlier in pixels

(in our tests between 0.8 and 1.5). Inlier with a higher error value are dropped. The second (INLIER_NUM) is the minimum number of inlier in a view. Views with a fewer number of inlier are dropped, too.

III EXTRACTING THE INTRINSIC PARAMETERS FOR INITIALIZATION

In general we write a calibration matrix in the following form:

$$K_{general} = \begin{bmatrix} \tau f & s & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

with f standing for the focal length, τ being the aspect ratio, s is the skew factor of the camera and (u_o,v_o) pointing out the principal point. Assuming that our skew factor is negligible we get the matrix

$$K = \begin{bmatrix} \tau f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6)$$

with only 4 degrees of freedom. Note that the whole calibration procedure could also be done including the skew factor and therefore using more homographies for parameter extraction. Furthermore projection is done by a projection matrix

$$sP = K[R|-t] \quad (7)$$

with R being a 3x3 orthogonal rotation matrix and t the translation vector, both together representing the 6 extrinsic parameters of the camera. The arbitrary scale factor is represented by s (to simplify matters it is replaced by the symbol '≃' in the following equations). Due to the fact that we use a planar calibration pattern we may transform (7) into

$$H \simeq KR \begin{bmatrix} 1 & 0 & -t_1 \\ 0 & 1 & -t_2 \\ 0 & 0 & -t_3 \end{bmatrix} \quad (8)$$

representing the homography mapping between our calibration object and the screen by dropping the column representing the Z-coordinate of our object. Calibration will be performed via the determination of the image of the Absolute Conic (IAC) ω:

$$\omega \simeq (KK^T)^{-1} = K^{-T}K^{-1} = \frac{1}{\tau^2 f^2} \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & \tau^2 & -\tau^2 v_0 \\ -u_0 & -\tau^2 v_0 & \tau^2 f^2 + u_0^2 + \tau^2 v_0^2 \end{bmatrix} \quad (9)$$

The AC is located in the plane at infinity and has the property to be (like its image IAC) invariant under Euclidean transformations. This makes it to a nice calibration object which is naturally present in every scene.
The calibration constraints arising from homographies can be expressed and implemented in several ways. For example, it follows from (8) that

$$H^T \omega H = H^T K^{-T} K^{-1} H \simeq (KRT)^T K^{-T} K^{-1} (KRT) = T^T R^T K^T K^{-T} K^{-1} K R T$$
$$= T^T R^T R T = T^T T = \begin{bmatrix} 1 & 0 & -t_1 \\ 0 & 1 & -t_2 \\ -t_1 & -t_2 & \|t\|^2 \end{bmatrix} \quad (10)$$

with the camera position t being unknown and holding up to scale factor only. Obviously we can extract the two following equations:

$$\begin{aligned} h_1^T \omega h_1 - h_2^T \omega h_2 = 0 \\ h_1^T \omega h_2 = 0 \end{aligned} \quad (11)$$

with h_i representing the *i* th column of homography H. Let us define the matrix

$$\Phi = \begin{bmatrix} 1 & 0 & \phi_{13} \\ 0 & \phi_{22} & \phi_{23} \\ \phi_{13} & \phi_{23} & \phi_{33} \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & \tau^2 & -\tau^2 v_0 \\ -u_0 & -\tau^2 v_0 & \tau^2 f^2 + u_0^2 + \tau^2 v_0^2 \end{bmatrix} \simeq \omega \quad (12)$$

and the solution vector x for our equation system Ax=b:

$$x = \begin{bmatrix} \phi_{13} & \phi_{22} & \phi_{23} & \phi_{33} \end{bmatrix}^T \quad (13)$$

Since $H = sH$ for every scalar s we obtain by solving (11) to the elements of (13) the following two rows for our problem Ax=b for each homography in the system:

$$\begin{bmatrix} 2(h_{11}h_{31} - h_{12}h_{32}) & h_{21}^2 - h_{22}^2 & 2(h_{21}h_{31} - h_{22}h_{32}) & h_{31}^2 - h_{32}^2 \\ h_{11}h_{32} + h_{12}h_{31} & h_{22}h_{21} & h_{32}h_{21} + h_{22}h_{31} & h_{32}h_{31} \end{bmatrix} \begin{bmatrix} \phi_{13} \\ \phi_{22} \\ \phi_{23} \\ \phi_{33} \end{bmatrix} = \begin{bmatrix} h_{11}^2 - h_{12}^2 \\ h_{11}h_{12} \end{bmatrix} \stackrel{\text{def}}{=} b \quad (14)$$

Thus, to determine equation (14), we need at least two homographies in contrast to the referenced approach [4] which makes use of at least three homographies. We may also add more homographies to increase accuracy and thus solve the following normal equation:

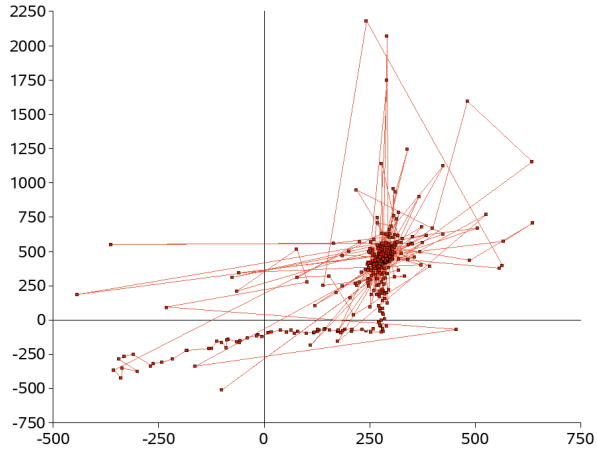
$$\begin{aligned} \|Ax - b\| = \min! \\ \Leftrightarrow A^T A x = A^T b \end{aligned} \quad (15)$$

For implementation issues this could be done by using the SVD $A^TA = U\Sigma V^T$ where $\Sigma = \text{diag}(\sigma_1,\sigma_2,\sigma_3)$ with σ_i singular values of A^TA and $U^TU = V^TV = I$. It follows from (12) that after having determined x, the intrinsic parameters are extracted via:

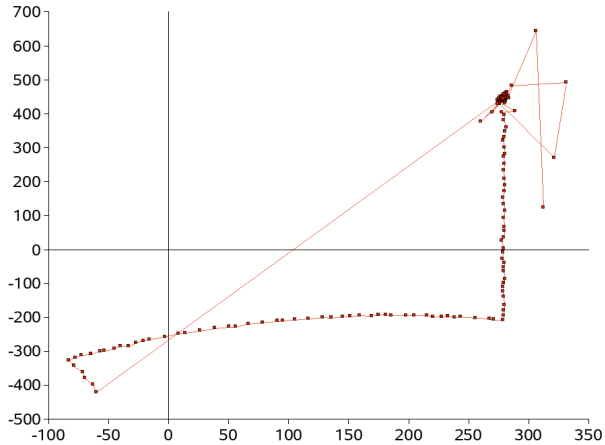
$$\tau^2 = \phi_{22} \quad u_0 = -\phi_{13} \quad v_0 = -\frac{\phi_{23}}{\phi_{22}}$$
$$f^2 = \frac{\phi_{22}\phi_{33} - \phi_{22}\phi_{13}^2 - \phi_{23}^2}{\phi_{22}^2} \qquad (16)$$

IV Experimental Results of Extracting the Intrinsic Parameters

This section is dedicated to the experimental results obtained using the estimation technique for the camera's intrinsic parameters which is described above. In each of the experiments an artificial calibration pattern computed from its rotation and translation was moved around on the screen using a predefined projection. Intrinsic parameters of the camera were: focal length = 1000, aspect ratio = 1, u_o = 475 and v_o = 275. Gaussian noise has been introduced using the Box Muller method [7]. After solving the normal equation (15), the resulting focal length, aspect ratio and principal point have been stored and plotted.
The following two figures depict a calibration plane which has been rotated with the help of the mouse (this is the reason for the arising linearities which appear close to "singular" positions of the plane). The pattern is captured from two views and Gaussian noise was introduced with σ = 1.0. The graphic on the left side was created using 4 feature points while the right one uses 1000 of them. It catches one's eyes that results are more precise using a higher number of feature points:



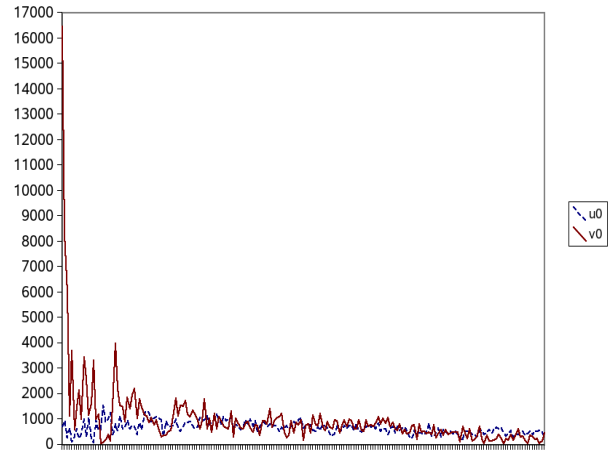
u_0 over v_0 using 4 feature points



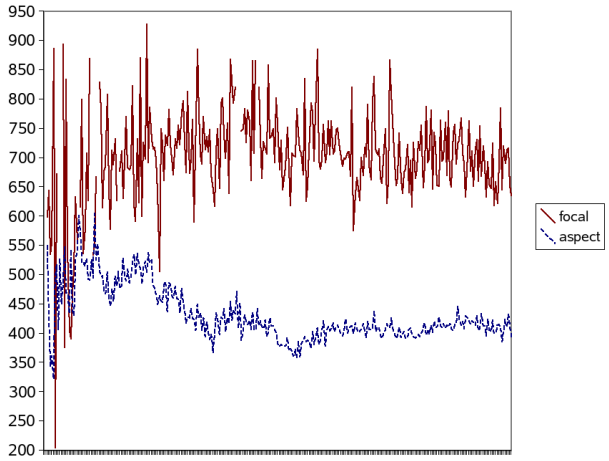
u_0 over v_0 using 1000 feature points

In the following, an experiment was made which shows the impact of increasing the number of views from 3 to 300 on the estimated parameters. 100 random feature points have been distributed uniformly over the calibration pattern using a noise of σ = 1.0. The error was measured absolutely in 1/1000 pixel for (u_o, v_o) and relatively in ‰ for focal length and aspect ratio. As can be seen clearly, the error of the principle point decreases up

to 100 views and begins to stagnate in the following while the error of focal length and aspect ratio doesn't encounter much influence:

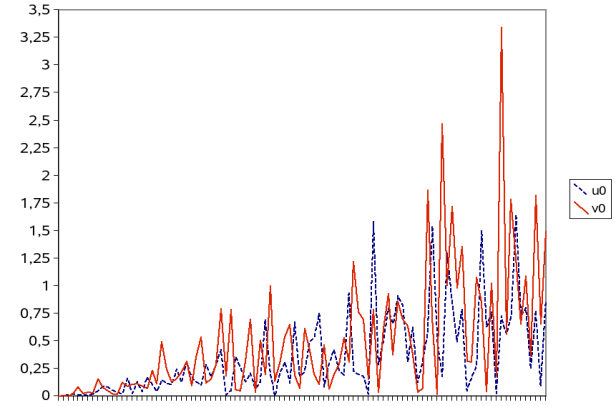


u_0 and v_0 error in 1/1000 pixels while increasing the number of views from 3 to 300.

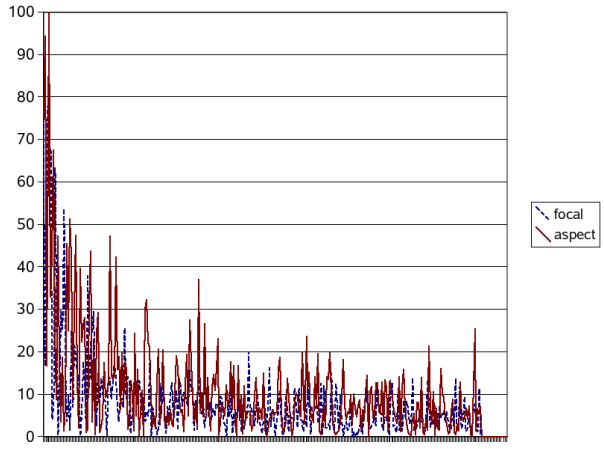


Focal and aspect error in ‰ while increasing the number of views from 3 to 300.

Another trial aimed at estimating the influence of increasing Gaussian error on the result. In the following figure on the left a calibration pattern with 4 feature points can be seen from 3 views. Noise has been linear increased up to $\sigma = 0.1$ while the resulting error was measured in pixels. With increasing noise the average error increases in a linear way. The figure on the right side shows the impact of increasing the number of feature points from 4 to 300 using 3 views and a noise of $\sigma = 0.1$. The resulting relative error is provided in ‰ again. This depicts very well that a higher number of feature points result in a more precise estimation of focal length and aspect ratio:



u_0 and v_0 error in pixels while increasing the noise linearly to 0.1.



Focal and aspect error in ‰ while increasing the number of feature points from 4 to 300.

V A Metric to Describe Homographies' Qualities

After calculating homographies like described in the first chapter, we obtain a set of homographies H_{ij} for every Camera i and every captured position j of the planar pattern. Let us define the matrix Ψ by

$$\Psi_{ij} = \begin{cases} 1 \text{ if homography from plane j to camera i exists} \\ 0 \text{ else} \end{cases} \quad (17)$$

Because of 4 dof we have to use at least two homographies matrices. It turned out that using all homographies available easily leads to singularities and complex results during the estimation process. Due to this fact we have to think of a filtering method to get rid of singularities. Nevertheless, for our global optimization in the end we may use these homographies again.

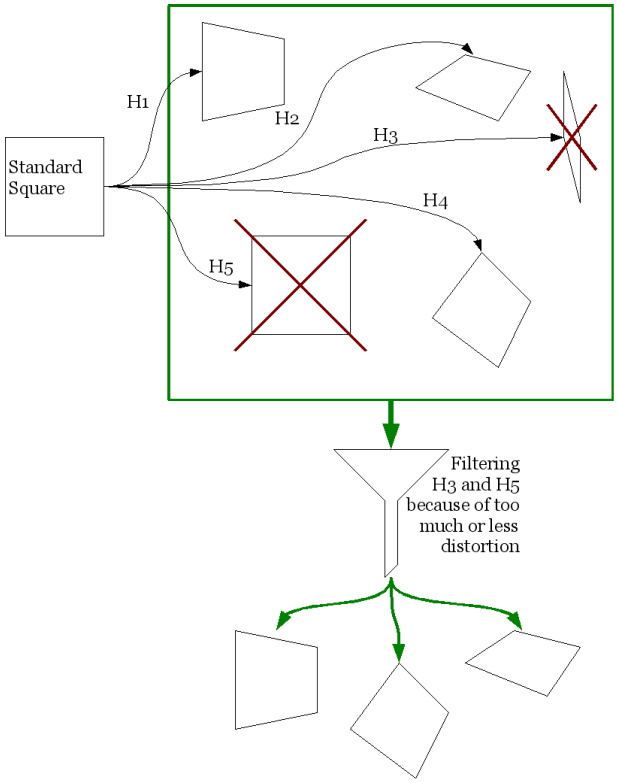


Illustration of the filtering process which takes into account homographies which distort the image too much or too less. This prevents singularities arising during the calibration process of the cameras.

In order to filter bad homography candidates from the set of homographies we introduce a metric allowing us to drop the bad ones in order to get a first guess of our calibration matrix. To prevent the homographies from being badly rotated we use a standard square S = {(0,0),(0,1),(1,0),(1,1)} and measure the angle in every corner of its projection PS

$$\forall_{x,y \in \text{edge}(PS)}: \quad \alpha < \operatorname{acos}\frac{x\,y}{|x||y|} < \frac{\pi}{2} - \beta \;\vee\; \frac{\pi}{2} + \beta < \operatorname{acos}\frac{x\,y}{|x||y|} < \pi - \alpha \quad (18)$$

using $0 < \alpha,\beta < 1$. Good results have been achieved by using $\alpha = 0.01$ and $\beta = 0.005$. In the same way the length ratio of the edges of the projected square PS is measured

$$\forall_{x \in \text{edge}(PS)}: |x| > \frac{\gamma}{4}\sum_{i=1}^{4}|v_i| \quad (19)$$

with v_i representing the edge vectors of PS. Experiments showed that a value of $\gamma = 0.9$ provides good filtering abilities.

Furthermore we thought of the possibility of using the following metric to describe the "distance" between 2 homographies

$$\max_{\mathbb{R}} \left\| \frac{H_1}{\|H_1\|_F^2} - \frac{H_2}{\|H_2\|_F^2} \right\|_F^2 \quad (20)$$

in order to select only two of them containing the most part of information. Since this brought no further improvements the idea was dropped in the following.
After filtering (18),(19) the parameters can be extracted and put into the calibration matrix K_i for camera i using (16). Once the calibration matrix for all cameras is estimated we may directly compute the corresponding rotation/translation matrices $[R|t]_i$ using

$$H = KR[u_1|u_2|-t] \;\Leftrightarrow\; R[u_1|u_2|-t] = K^{-1}H = \begin{bmatrix} \frac{1}{\tau f} & 0 & -\frac{u_0}{\tau f} \\ 0 & \frac{1}{f} & -\frac{v_0}{f} \\ 0 & 0 & 1 \end{bmatrix} H \quad (21)$$

with u_x being the x-th unit vector. We may complete the missing column of R (in the following called Q since it is not yet orthonormal) by calculating the orthogonal vector to r_1 and r_2. Finally we get $Q = [r_1|r_2|r_3]$ and the translation vector t with

$$\begin{aligned} r_1 &= \lambda K^{-1} h_1 \\ r_2 &= \lambda K^{-1} h_2 \\ r_3 &= r_1 \times r_2 \\ t &= \lambda K^{-1} h_3 \end{aligned} \quad (22)$$

with $\lambda = 1/|K^{-1}h_1| = 1/|K^{-1}h_2|$ for normalization. Unfortunately Q does not in general satisfy the properties of a rotation matrix. Therefore the next orthonormal matrix R is searched with the smallest distance to Q in the Frobenius sense. We may express this by

$$\min_{R} \|R-Q\|_F^2 \quad (23)$$

with

$$\begin{aligned} \|R-Q\|_F^2 &= trace((R-Q)^T(R-Q)) \\ &= trace(R^T R) + trace(Q^T Q) - 2\ trace(R^T Q) \\ &= 3 + trace(Q^T Q) - 2\ trace(R^T Q) \end{aligned} \quad (24)$$

By considering $trace(Q^TQ)$ as constant, (23) equals to the maximizing task of $trace(R^TQ)$. Let the singular value decomposition of Q be $U\Sigma V^T$, where $\Sigma = diag(\sigma_1,\sigma_2,\sigma_3)$ with σ_i singular values of Q and $U^TU = V^TV = I$. Now we define the orthogonal matrix Z by $Z = V^TR^TU$. Then there is:

$$\begin{aligned} trace(R^T Q) &= trace(R^T U \Sigma V^T) \\ &= trace(V^T R^T U \Sigma) \\ &= trace(Z\Sigma) \\ &= \sum_{i=1}^{3} z_{ii}\sigma_i \;\le\; \sum_{i=1}^{3} \sigma_i \end{aligned} \quad (25)$$

The maximum is achieved by setting R = UV^T because in this case Z equals the identity matrix and the trace of R^TQ reaches the maximal value $\sigma_1+\sigma_2+\sigma_3$. This provides us with the set of matrices $[R|-t]_{ij}$, each corresponding to the homography H_{ij}.

VI Global Optimization using Levenberg-Marquardt

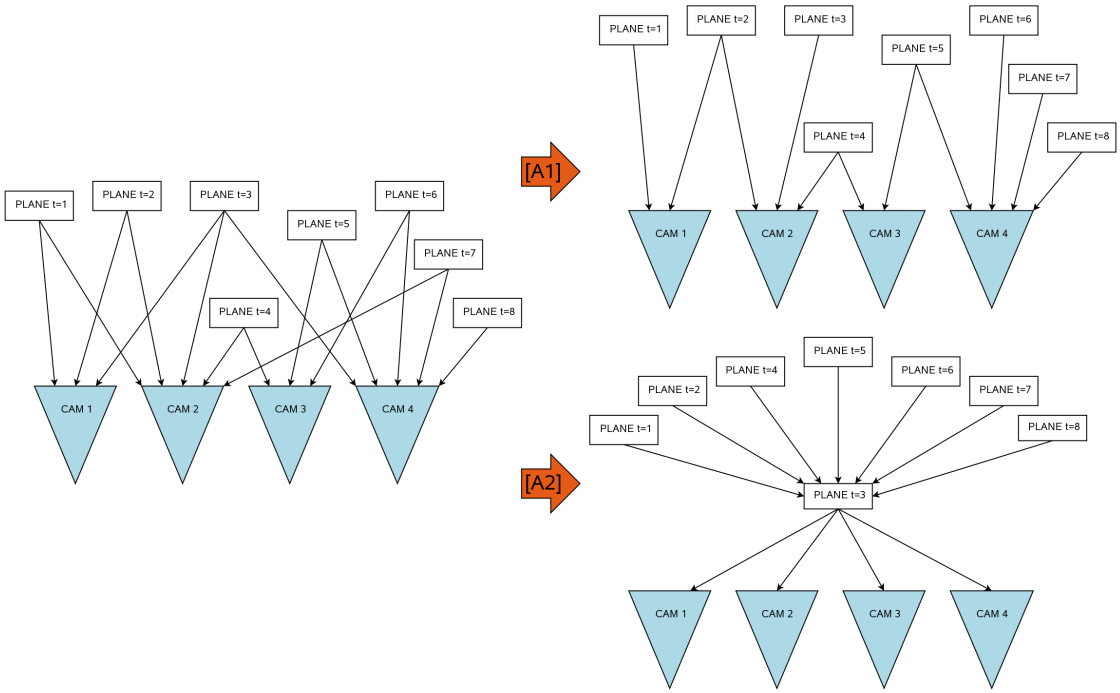
Using the results from the previous chapter we finally get the following matrices:

$$\begin{matrix} \forall_{i=1..n}: K_i \\ \forall_{i=1..n}: \forall_{j=1..m \text{ with existing } H_{ij}}: RT_{ij} \end{matrix} \quad (26)$$

This leads us to the following question: Are there enough matrices RT_{ij} to connect all cameras? In other words:

$$\exists_{j^0 \in \{1..m\}}: \forall_{i=1..n}: \exists_{A \subset (\{1..n\} \times \{1..m\})}: \left[\forall_{a \in A}: \exists: RT_a \quad \wedge \quad RT_{ij^0} = RT_{a_1} (RT_{a_2})^{-1} RT_{a_3} \ldots RT_{a_k} \right] \quad (27)$$

In order to do global optimization we have to make sure that there arise no ambiguous links. Therefore we have to discard all redundant connections which can be reproduced by a chain of other connections. An algorithm [A1] was developed for this task allowing fast reduction of the matrix Ψ. After executing [A1] each discarded connection can be reproduced by a chain of matrices. Because the Levenberg-Marquardt algorithm needs the gradient, the use of more than 2 matrices for projection turned out to be suboptimal. Therefore a new algorithm [A2] was designed. It tries to find the plane j^0 with the maximal connection number to the cameras and calculates the [R|-t] matrices of all other planes to j^0 as well as all [R|-t] matrices from j^0 to each of the camera. An example of the algorithms' transformations is shown in the following graphs:



[A2] can be formulated in short pseudo code:

```
Find plane j with maximal number of existing homographies
Find camera i with maximal number of existing homographies

DO
 FOR All x as camera DO
  FOR All y as plane DO
   IF Ψxy = 0 THEN
        Try to recover connection
WHILE at least one connection has been recovered

IF all cameras and enough planes are connected THEN
 RETURN TRUE
```

In the case that [A2] returns false we do not have enough views to calibrate all the cameras. Otherwise we receive a connection matrix Ψ_{new} like illustrated in the following example for 4 cameras and 5 views

$$\Psi_{old} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \underset{[A2]}{\Rightarrow} \Psi_{new} = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 1 & 1 & \mathbf{1} & 1 & 1 \\ 1 & 1 & \mathbf{1} & 1 & 1 \\ 1 & 1 & \mathbf{1} & 1 & 1 \end{bmatrix} \quad (28)$$

with the selected maximum column and row marked bold. In order to perform our final global optimization we have to minimize

$$\sum_{\underbrace{i=1}_{Cam}}^{n} \sum_{\underbrace{j=1}_{View}}^{m} \sum_{\underbrace{k=1}_{Point}}^{p} \left\| \begin{pmatrix} u_k \\ v_k \\ 1 \end{pmatrix}_{ij} - H_{norm} \left[K_i [R|-t]_i [R|-t]_j^{-1} \begin{pmatrix} x_k \\ y_k \\ 0 \\ 1 \end{pmatrix} \right] \right\|_2^2 \quad (29)$$

in all $6((m-1)+n)+4n$ dimensions with H_{norm}(vector) representing the normed vector in homogeneous coordinates sense. Note that not in all views the calibration object might be seen and therefore these one have to be excluded from the optimization process using Ψ_{old} for their determination. The matrices K_i and $[R|-t]_i$ may be directly taken from the connections represented by Ψ_{new} while we may want to preprocess the calculation of the inverse matrices $[R|-t]_j^{-1}$. These inverse matrices are directly computed to

$$[R|-t]_j^{-1} = \begin{bmatrix} R^T & R^T t \\ 0 & 1 \end{bmatrix} \quad (30)$$

since

$$R^{-1} = R^T \quad \wedge \quad \begin{bmatrix} R^T & R^T t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & -t \\ 0 & 1 \end{bmatrix} = I \quad (31) .$$

Being a non-linear minimization problem we make use of the Levenberg-Marquardt algorithm [6] for solving this system.

VII Pre-Filtering

In a normal calibration environment feature point detection may provide us with too many views to optimize on. In our experiments it turned out that 200 views lead to reliable and good results while taking the optimizer a time of about 3-4 minutes on a 3.0 Ghz Pentium IV machine. For this reason we suggest to do a primary filtering process on the data in order to obtain a set of valuable homographies. The following three filter types have been implemented:

A Greedy Filter

Loops through the cameras of all views and grabs the homographies until Homography_Num (maximal number of homographies to use for optimization) is reached.

B Padding Filter

This filter owns an additional parameter, the Padding_Ratio. It implies 2 phases: in the first phase Homography_Num/(Camera_Num*Padding_Ratio) homographies for each of the cameras are extracted to ensure enough calibration data for each of the cameras. In a second phase the remaining amount of homographies is filled with the views endowing the most of the connections in a descending order. This ensures good connectivity which turned out to be very important for the system.

C Probabilistic Filter

The third filter which was implemented is based on a fitness function which composes to

$$Q(S_i)=w_p q_p(S_i)+w_c q_c(S_i)+w_d q_d(S_i) \quad (32)$$

with S_i being the i-th solution, w_x standing for some adjustable weights (points, connections, distance),

$$q_p(S_i)=\frac{\sum_j 1_{\text{point } j \text{ is inlier of } S_i}}{\max_i \sum_j 1_{\text{point } j \text{ is inlier of } S_i}} \quad (33)$$

specifying the number of points used in this solution,

$$q_c(S_i)=\frac{\sum_i (connection_{number}(homography_i))^2}{view_{number}\, camera_{number}} \quad (34)$$

marking the quality of connectivity of the solution and finally

$$q_d(S_i) = \frac{\sum_i \sum_{(H_1,H_2)\in\{(a,b):a,b\in Hom(Cam_i)\}} \|H_1 p - H_2 p\|_2}{\sqrt{image_{width}^2 + image_{height}^2} \sum_i |\{(a,b):a,b\in Hom(Cam_i)\}|} \quad (35)$$

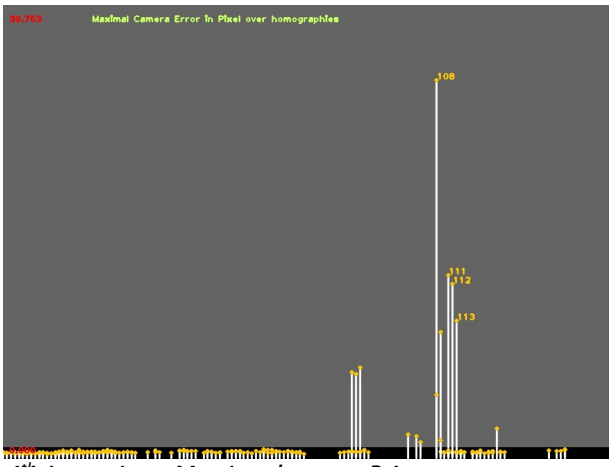
adding a distance metric to the quality function with p being the point (0,0,1). Afterwards a random algorithm chooses a set of solutions S_i and evaluates the fitness function for each of them to get them maximal value. Nevertheless it is only a small area of our parameter space which can be considered since the number of possible solutions is

$$\binom{Homographies\ available}{Homography_{Num}} \quad \text{and} \quad \binom{400}{200} \text{ exceeds } 10^{119} \text{ already, for instance.}$$

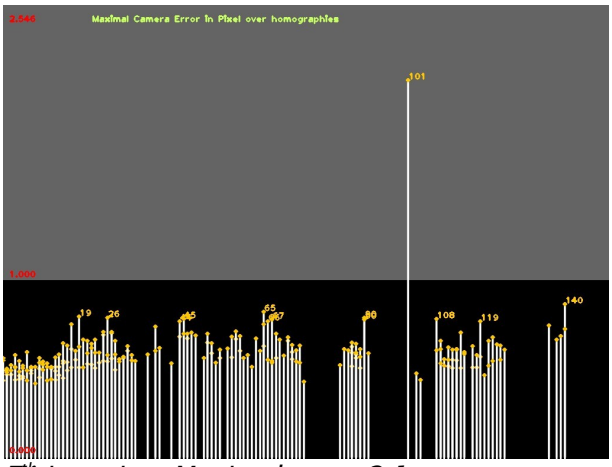
It turned out that filter B provides us with best results up to now, since filter A doesn't make use of qualitative information and filter C is too slow to cover a sufficiently large area of the parameter space.

VIII Post-Filtering

While running the global optimization process we could still observe ambiguities in single homographies which do not dissolve and could not be filtered before. The following figures depict the average error in pixels of each homography on the image plane after 4 (left) and 7 (right) iterations. The black region delimits an averaged error of 1 pixel:

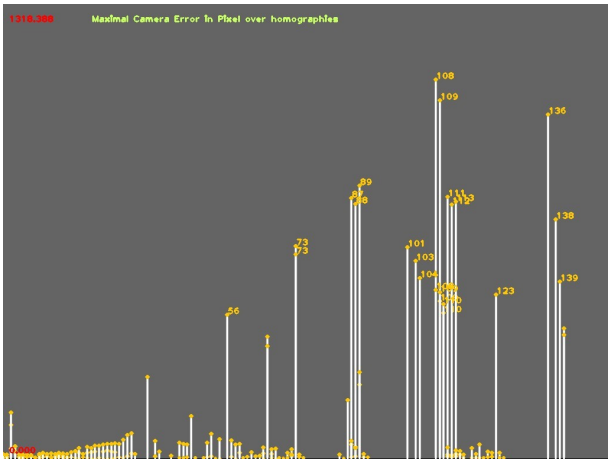


4th iteration: Maximal error 34

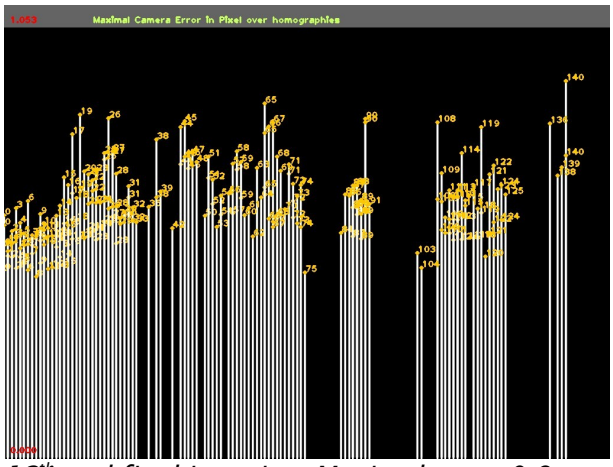


7th iteration: Maximal error 2.1

To eliminate remaining outliers we introduced a threshold τ (τ = 1.0) to define the maximal resulting error in pixels. After a certain number of iterations or if a certain accuracy has been reached, all outliers with an error > τ are removed and the optimization process continues. The left figure depicts our initial error histogram (0 iterations, maximal error about 1000 pixels) while the right one shows our final optimization results after 12 iterations with a maximal error of about 0.8 pixels:



0 iterations: Maximal error 1000 (!)

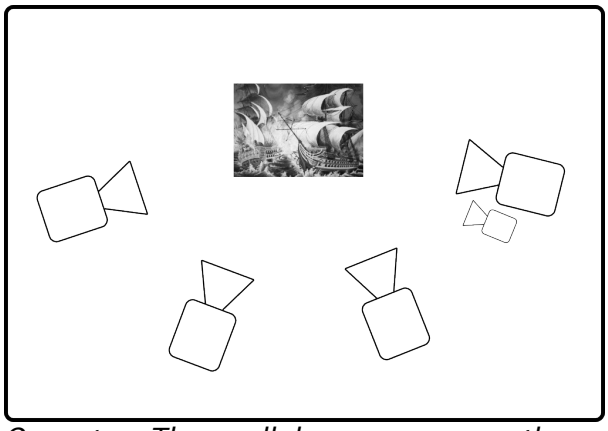


12th and final iteration: Maximal error 0.8

IX Calibration results

For our real hardware tests we made use of of 5 Basler Firewire cameras with a resolution of 1024x768. Acquiring software was Streampix III© of NorPix. Shutter frequency has been set to 5 Hz. Our calibration pattern we used and our test setup is depict on the following figures:

The calibration pattern which was stuck on a cardboard box: Bataille navale

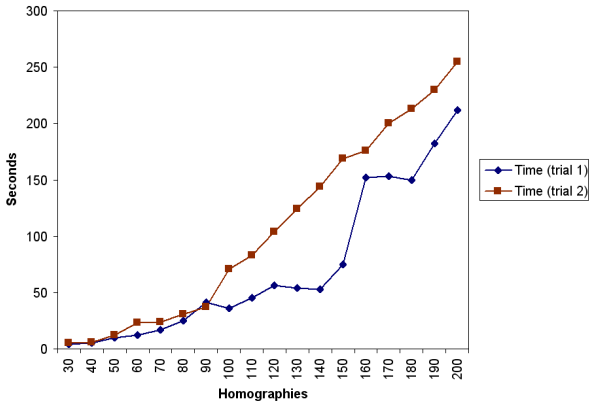


Our setup. The small drawn camera on the right side was below its neighbor.

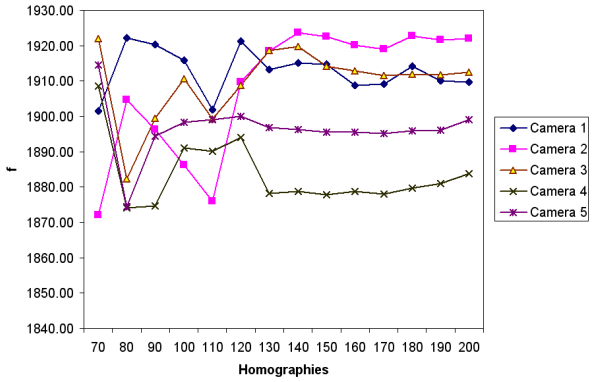
The calibration pattern had been stuck on a cardboard box and was presented to the synchronized cameras with different translations and rotations for about one minute. The way we presented the pattern to the cameras turned out to be crucial: Best calibration results have been achieved using many different rotation positions to cover the most part of the rotation space. Important was also to show the pattern to more than one camera at the same time which ensures good connectivity for our final optimization. The following picture shows our calibration environment:

Presentation of the calibration pattern in CVLAB at EPFL.

Two sequences have been taken: The first trial included 150 views for each of the 5 cameras while in the second one we took 270 images each. We used the parameters 50 (INLIER_MIN) and 1.2 (DISTANCE_MAX) while detecting feature points.
The maximal number of Levenberg-Marquardt iterations in one loop (OPT_ITERATIONS) was bounded to 10 (after realizing that the system converges really fast) and OPT_EPSILON was set to 10^{-4}. Finally we increased the number of homographies which were used for the optimization (HOMOGRAPHY_NUM) slowly to a maximum of 200 homographies which implies an optimization in a 1244 (!) dimensional parameter space. Concerning the time needed on a 3.0 Ghz machine we got the results depict on the left figure below. The right figure shows the evolution of the final parameter f (focal length * aspect ratio) over the number of homographies used for trial 1:

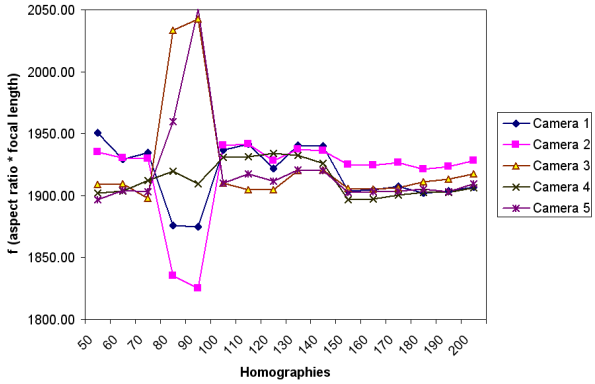


Evolution of time over number of homographies which were used.

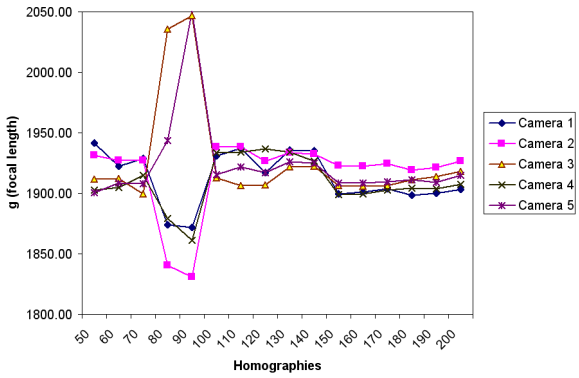


Evolution of parameter f in trial 1.

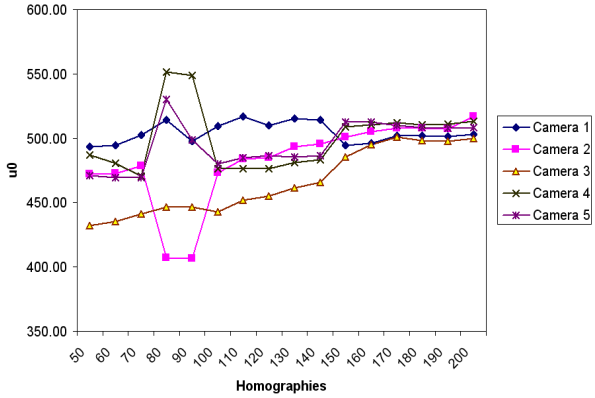
As can be seen time increases close to linearly with the number of homographies. Using more than 130 homographies leads to a stabilization process of parameter f. Unfortunately results for trial 1 were not very representative since we realized that there had been taken only very few images which connect camera 3 to the rest of the cameras. For this reason we proceed with the results of trial 2. The following set of figures illuminates evolution of each of the camera's intrinsic parameters over the ascending number of homographies used for the calibration procedure:



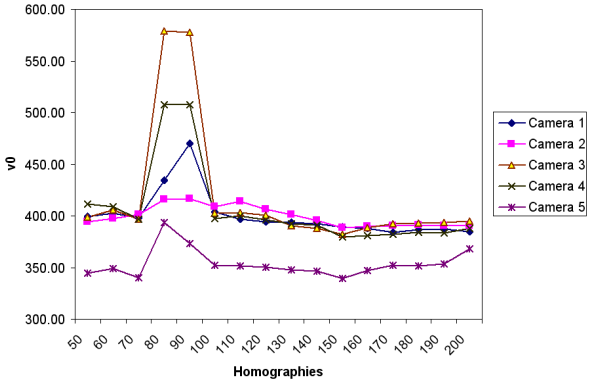
Evolution of f over number of homographies which were used in trial 2.



Evolution of g over number of homographies which were used in trial 2.



Evolution of u_0 over number of homographies which were used in trial 2.



Evolution of v_0 over number of homographies which were used in trial 2.

For each parameter a stabilization effect can be observed using more than 150 homographies. This suggests that our results come close to reality. Now let's have a closer look at our final result making use of 200 homographies for trial 2. Our calibration tool produces the following final output after having performed post-filtering and done 5 additional iterations (OPT_EPSILON was reached after 8 iterations in the first loop):

	Focal	Aspect	u_0	v_0
Camera 00:	1910.733	1.001	491.259	394.602
Camera 01:	1923.384	1.001	509.848	385.584
Camera 02:	1913.524	1.000	502.182	390.652
Camera 03:	1911.224	0.999	512.494	381.821
Camera 04:	1908.050	0.997	514.731	361.395

	Optimal	Original
Maximal error in observations:	4.647	1.639
Minimal error in observations:	0.004	0.004
Average error in observations:	0.522	0.489
Chi^2 error in observations:	0.185	0.179
Standard deviation of error:	0.311	0.261
Number of observations:	18706	18706

Errors are measured in pixels on the image plane. The “Original”-measurement means using the homographies directly coming from feature point detection as a 2D-2D projection which covers ambiguities, of course but serves as a kind of “ground truth” in order to compare our results. We may easily notice that our results are pretty good: We realize the homogeneity of the camera's intrinsic parameters as well as the proximity to the best possible results of our artificial ground truth. Note that similar error results are observed while calibrating the “Stanford Multi-Camera Array” [13]. To illustrate the quality of our algorithm we projected a bounding box on top of our calibration cardboard using the projection matrices which have been returned. The following screen shot of our video sequence shows the accurate geometric representation of the object in the scene:

Projected bounding box of our calibration object, seen from the 5 different cameras' views.

X Implementation

Implementation was done using Standard C++. The class CamCalibration has been created. It contains a small set of public functions as an Interface to the class. Such are AddCamera(), AddHomography(), Calibrate() and some functions to store results and statistics. In addition a simple demo framework has been written in Visual Studio .NET to demonstrate the usage of the class. It makes use of the PlanarObjectRecognizer [1],[2] for feature point detection and a parser written by Julien Pilet in order to parse a ini-file which contains the parameters. The whole project was documented using the Doxygen syntax [8].

XI Parameters

In this chapter a short overview summarizes the tunable parameters for our algorithm:

Parameter	Short description
CCP.STRING.REFERENCE	Reference path- and filename.
CCP.STRING.FILENAME	Path- and filename of image sequences.
CCP.INT.CAMERA_START	First and last number of the camera in the image sequences. Will be inserted into the filename.
CCP.INT.CAMERA_END	
CCP.INT.IMAGE_START	First and last number of the image in the sequences. Will be inserted into the filename.
CCP.INT.IMAGE_END	
CCP.DOUBLE.DISTANCE_MAX	Maximal distance (px) on screen during detection.
CCP.INT.INLIER_MIN	Minimal number of inlier during detection.
CCP.INT.HOMOGRAPHY_NUM	Number of homographies to use (rest is filtered).
CCP.INT.PRE_FILTER	Filter to use (Greedy, Padding, Probabilistic).
CCP.DOUBLE.PRE_FILTER_A	Parameters to tune the Padding (a=ratio) and the Probabilistic filter (a, b, c represent weights for the quality functions).
CCP.DOUBLE.PRE_FILTER_B	
CCP.DOUBLE.PRE_FILTER_C	
CCP.DOUBLE.INITIAL_GUESS_A	Parameters for initial intrinsic camera calibration guess: a and b are angles, c is a distance ratio according to the description in chapter 7.
CCP.DOUBLE.INITIAL_GUESS_B	
CCP.DOUBLE.INITIAL_GUESS_C	
CCP.INT.OPT_ITERATIONS	Iteration number for optimization during one loop.
CCP.DOUBLE.OPT_EPSILON	Accuracy of optimization.
CCP.DOUBLE.POST_FILTER	Post filtering: Maximal euclidean error distance.

A detailed description can be found in the parameters.ini template file!

XII Conclusion & Outlook

In this work we tried to calibrate multiple cameras based on sequences of synchronized taken pictures of a planar calibration pattern. We were faced with problems of singularities while doing our initial guess which we solved using the projection of a planar square. We offered filtering possibilities to reduce information to a tractable number. Finally we evaluated our algorithm concerning time and quality aspects and now provide a class which is easy to use and to customize and might serve as a base for further projects.

Nevertheless more research in this field has to be done to allow more precise calibration of multiple cameras. A high number of dimensions for global optimization makes the problem intractable regarding temporal aspects. Due to this a good pre-filtering method has to be developed which generates a set of homographies containing the most part of information in the sequences. Furthermore radial distortion was not considered in this approach and has to be tackled, too.
Connecting these results with the simultaneous estimation of light sources could be useful in the field of augmented reality. The next step we will do is trying to establish this connection using Lambertian reflectance or an additional mirror attached to the calibration pattern.

XIII Acknowledgment

I would like to thank Julien Pilet and Vincent Lepetit for attending me in the course of my project and answering all of the questions I had. Furthermore I would like to thank Prof. Pascal Fua, leader of the Computer Vision Group at EPFL, who made this project possible and admit me to work for 2 additional weeks in the Computer Vision Laboratory.

XIV References

[1] V. Lepetit, P. Lagger, P. Fua. *Randomized Trees for Real-Time Keypoint Recognition*. Conference on Computer Vision and Pattern Recognition, San Diego, 2005.
[2] V. Lepetit, P. Fua. *Towards Recognizing Feature Points using Classification Trees*. Technical Report IC/2004/74. Ecole Polytechnique Fédérale de Lausanne, 2004.
[3] Zhengyou Zhang. *A Flexible New Technique for Camera Calibration*. Technical Report MSR-TR-98-71, Microsoft Research, 1998.
[4] Peter F. Sturm, Stephen J. Maybank. *On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications*. IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, USA, p. 432-437.
[5] R. Hartley, A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004.
[6] W. Press, S. Teukosky et al. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 2002.
[7] *Wikipedia 2005*. *http://en.wikipedia.org/wiki/Box_muller*.
[8] D. v. Heesch. *Doxygen documentation system*. *http://www.doxygen.org/*.
[9] R. Tsai. *A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses*. IEEE Journal of robotics and automation, Vol. A-3, No. 4, August 1987.
[10] Aneliya Mircheva. *Kamerakalibration*. Seminar: Robotik und Medizin. University of Karlsruhe (TH), 2004.
[11] Y. Abdel-Aziz, H. Karara. *Direct linear transformation into object space coordinates in close-range photogrammetry*. Proc. Symp. Close-Range Photogrammetry. University of Illinois at Urbana-Champaign, Urbana, 1971.
[12] *The Stanford Multi Camera Array*. *http://graphics.stanford.edu/projects/array/*
[13] *The Stanford Multi Camera Array*. *http://graphics.stanford.edu/~vaibhav/projects/lfca_calib/*