Self-Calibration of a Pair of Webcams for Stereo Vision

Rebecca Illowsky and Landry Huet
Stanford University
CS 221 and CS 229 final project

December 14, 2007

1 Motivations

The idea of the project was originally given by Steve Gould, a PhD student at the AI laboratory. Researchers often waste a lot of time calibrating the cameras they are using as sensors for their robots. There are two kinds of parameters for cameras. The first ones are internal to the cameras. They account for the properties of their lenses and for their defects. Using these parameters helps getting rid of both linear and quadratic deformations of the pictures and therefore enhance the pictures for use with computer vision algorithms. They are called the *intrinsic parameters* because they depend only on the cameras themselves and not on their environment. They only need to be computed once. The second kind of parameters is the class of *extrinsic parameters.* They relate the cameras to the outside world. Usually the only interesting extrinsic parameters are the positions and orientations of the cameras. In the case of a pair of cameras, the extrinsic parameters are essential for stereoscopic vision. However, these parameters are sensitive to the outside world. If an engineer is tuning a robot, it means that he will have to start the calibration of extrinsic parameters all over again every time the cameras of his robot are moved.

A common technique for camera calibration requires taking multiple pictures of a checkerboard under various angles, and then feeding them to the *matlab camera calibration toolbox.* Although the process is effective, it is very tedious and takes at least half an hour for a trained operator. This is not what you want to focus on when you are working on a robot, but it might use a lot of your time and energy nonetheless if you have to remove the video sensors of the robot and put them back often. Our aim was therefore to design a technique that would make the robot recalibrate the extrinsic parameters of its video sensors by itsself, just by looking at the outside world.

2 Theoretical background

2.1 Scene reconstruction using two cameras

Let $(\mathcal{C}_1, \mathcal{C}_2)$ be a pair of cameras. Let us suppose that we know the translation vector $\overrightarrow{t}$ and the rotation matrix R that link the image plane of camera $\mathcal{C}_1$ to camera $\mathcal{C}_2$, as well as the matrices A_1 and A_2 accounting for the intrinsic parameters of each camera. Let $\overrightarrow{M} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$ be the position vector of a physical point M in space, relative to the frame of the first camera. Let P_1 and P_2 be the projections of M on the frames of cameras $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively, and $\widetilde{P}_1 = \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}$ and $\widetilde{P}_2 = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix}$ be their homogeneous coordinates in their respective frames. M lies on the lines linking the focal centers of the cameras $\mathcal{C}_1$ and $\mathcal{C}_2$ to P_1 and P_2 respectively, so there must exist two real numbers s_1 and s_2 such that:

$$\begin{cases} \overrightarrow{M} & = & s_1 A_1^{-1} \widetilde{P}_1 \\ \overrightarrow{M} & = & s_2 R^\top A_2^{-1} \widetilde{P}_2 - \overrightarrow{t} \end{cases}$$

Except in degenerate cases there is always a unique solution to this system of equations. This is not the case when doing numerical computation but $\overrightarrow{M}$ can be well approximated by plugging the result of the minimization problem

$\min_{(s_1,s_2)} \left\| s_1 A_1^{-1} \widetilde{P}_1 - s_2 R^\top A_2^{-1} \widetilde{P}_2 + \overrightarrow{t} \right\|_2$ back into the system. Hence the need for R and $\overrightarrow{t}$ to reconstruct the position of points in space.

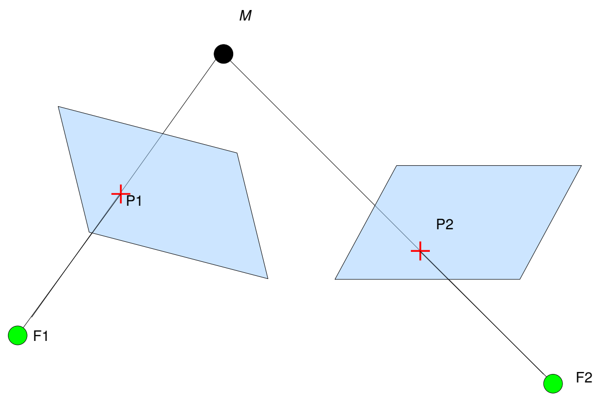
2.2 General scheme to compute the extrinsic parameters

The first step of our method is to find corresponding points in various pairs of images. It turns out that for our purpose it is not necessary that all the points belong to the same pair of images. From these correspondences we can infer the epipolar constraint in the form of a matrix. The epipolar constraint is the fact that all the points of the image plane of $\mathcal{C}_2$ that could possibly be related to a given point P_1 of the image plane of $\mathcal{C}_1$ as coming from the same physical point in space lay on a straight line. The knowledge of this constraint then gives us the translation vector $\overrightarrow{t}$ between the origin of the image plane of camera $\mathcal{C}_1$ and that of $\mathcal{C}_2$, as well as the rotation R of space that transforms the first image plane into the second image plane.

We started from pairs of corresponding points. We did not want to study this part because there are already plenty of librairies that can do the job, and because it should really be up to the user of our method to choose which one he prefers. Let us now take a closer look at each step of our algorithm.

2.3 Getting the essential matrix

Using the same notations as in subsection (2.1), let us consider a point M that projects onto P_1 and P_2. Let us moreover consider F_1 and F_2 the respective focal centers of cameras $\mathcal{C}_1$ and $\mathcal{C}_2$.



P_1, P_2 and M are coplanar. With respect to an arbitrary origin this constraint yields the equation :

$$P_1^\top \left(\overrightarrow{t} \times (RP_2) \right) = 0$$

This is bilinear in P_1 and P_2 and can be summed up by the matrix $E = \widehat{T}R$ where $\widehat{T}$ is the matrix of the map $\overrightarrow{u} \mapsto \overrightarrow{t} \times \overrightarrow{u}$. E is called the *essential matrix* of the pair of views, and it sums up all the geometry of the system. In practice the 3D coordinates of P_1 and P_2 are not directly known. We can fix this by considering $\widetilde{P}_1 = A_1 P_1$ and $\widetilde{P}_2 = A_2 P_2$, as defined in subsection (2.1). Writing the same constraint now gives us

$$\widetilde{P}_1^\top A_1^{-\top} \widehat{T} R A_2^{-1} \widetilde{P}_2 = 0 \tag{1}$$

$F = A_1^{-\top} \widehat{T} R A_2^{-1}$ is called the *fundamental matrix* of the uncalibrated system. Since $\widetilde{P}_1$ and $\widetilde{P}_2$ can be directly measured, we can use (1) to compute F and then infer E, given a sufficient number of reliable corresponding points between our two frames. F and E are determined up to a scale factor, so since we want to keep a track of the sense of displacement, if we know a non-null essential matrix E, we will also be interested by $-E$.

2.4 From the essential matrix to the extrinsic parameters

Given $E = \widehat{T}R$ we are looking for $\widehat{T}$ and R. According to ([1]), if $E = U\Sigma V^\top$ is the SVD of E and $R_z = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ then there are exactly two possible pairs of values for $\widehat{T}$ and R :

$$\begin{aligned} \left(\widehat{T}_1, R_1\right) &= \left(U R_z \Sigma U^\top, U R_z^\top V^\top\right) \\ \left(\widehat{T}_2, R_2\right) &= \left(U R_z^\top \Sigma U^\top, U R_z V^\top\right) \end{aligned} \tag{2}$$

Since we get two essential matrices already, we now have four possible solutions, three of which are physically incorrect. It is possible to find out by reconstructing the spatial positions linked to the pairs of corresponding points we already used for the calibration. Then we can simply discard the pairs $(\widehat{T}, R)$ that reconstruct one or more points behind the cameras (negative z-value). Note that since the essential matrix is defined up to a scale factor, so is the translation vector we find. Theory tells us this is the best we can do without a measurement of a recognized object in the actual scene.

3 From theory to practice

3.1 Implementation

We programmed a *C++* implementation of the method.

Point correspondences We used the Scale-Invariant Feature Transform (SIFT) algorithm from the STAIR research group, which seems to give fairly acurate results.

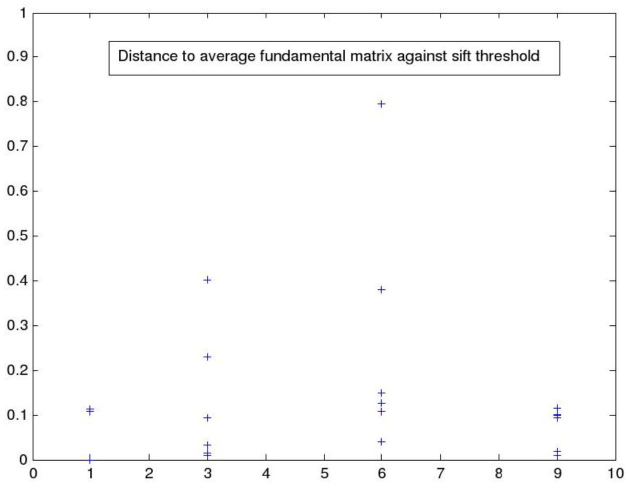
Essential matrix The *Open Computer Vision* library has a built-in function that can compute the fundamental matrix of a pair of pictures given pairs of correspondences. Our program gives it all the correspondences the first step could find. The essential matrix is then obtained with two matrix multiplications.

Retrieving the extrinsic parameters We use the SVD decomposition function from the *Open Computer Vision* library. It is impossible to select the correct parameters just by eliminating the parameters that produce bad reconstructions, because incorrect correspondences may give inconsistent reconstructions even with the right parameters. We therefore keep the solution that produces the least bad results. We assume that this will be correct if we have many more good correspondences than bad correspondences.

3.2 Results

In order to assess our method we conducted a series of 21 experiments[1]. All the experiments were run on the same setup of our pair of cameras. There were four runs of experiments, three with an incresing threshold for SIFT (therefore getting more and more correspondences of decreasing quality), the last one with another technique (see [3]) to replace SIFT. For each run we used more and more pictures and therefore got more and more points.

Fundamental matrix Since our method is supposed unbiaised, the mean of the fundamental matrices we get can be considered the closest guess we have about the true fondamental matrix of our system. Let us plot the distance to this best guess against the SIFT threshold (1 stands for [3]):

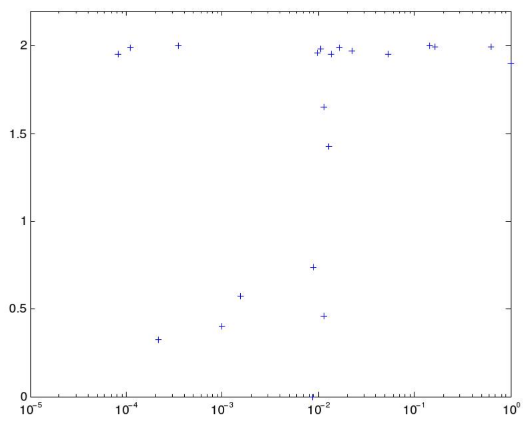


The leftmost points refer to the experiments with the best correspondences but the least points, whereas the rightmost points correspond to the experiments with the most correspondences, of lesser quality. We can infer from this that quantity and quality are about equally acceptable to find a good fundamental matrix.

The final extrinsic parameters only depend on the fundamental matrix. It is likely that better fundamental matrices will give better results in the end, so let us choose a particularly successful experiment as far as the fundamental matrix is concerned, and assess the quality of the other experiments by comparing their parameters to that of our selection. Let us plot the distance between the translation vectors[2] and our supposedly good translation vector, against the quality of the fundamental matrix (distance to the average):

[1]You can email rebeccai@stanford.edu for full results.

[2]The translation vectors are normalized so this quantity is to be considered with care, all the more as the vectors we found are scattered. A similar problem would arise with the rotations.



As we can see, even if the supposed quality of the fundamental matrix is good, the final result is not necessarily acurate. However, as soon as the fundamental matrix goes bad, the result for the translation vector is always very bad. The bad conditionning of the matrices of intrinsic parameters is probably accountable for the extreme sensitivity of the end result to inaccuracy of the fundamental matrix.

Conclusion

The results of our work are promising but our implementation should be tuned up before being used in real applications. Even though it is doubtful that self calibration will be as acurate as the *matlab calibration toolbox*, our technique can be very helpful when time is a bigger issue than precision, as is sometimes the case in experimental robotics.

Finally our work could be improved in several ways:

- By improving accuracy, for instance with a better algorithm to find correspondences or to approximate the fundamental matrix of the system.
- The program could be made more iterative, so that the user could give the algorithm correspondences until a certain degree of precision is reached.

Bibliography and acknowledgements

References

[1] Yi Ma, Stefano Soatto, Jana Košecká, S. Shankar Sastry: *An Invitation to 3-D Vision*

[2] Quang-Tuan Luong, Olivier Faugeras: *Self-calibration of a stereo rig from unknown camera motions and point correspondences*

[3] MetaMorph V 6.2r6 Universal Image Corp. Automated object tracking with "Track Objects" App

Best thanks to the STAIR research group and to Steve Gould in particular.