Calibration of CCD cameras for computer aided surgery

Krisztina Maróti*

Department of Image Processing and Computer Graphics
University of Szeged
Szeged / Hungary

Abstract

A navigation system is planned and implemented in order to use it in computer aided surgery. The idea is that three cameras are collecting images of the same object and from these projections the 3D coordinates of the points can be computed. In order to perform such a positioning we have to solve the calibration of the cameras. The calibration needs a special object, called calibration cross. The images of the calibrated cameras can be used later for determining point positions. The experimental results shows that the positions of 3D points can be determined with an error cca. 0.3 cm. We work further for improving this result.

Keywords: camera calibration, computer aided surgery, Powell's algorithm

1 Introduction

Skeletal injury operations are in general of high complexity and require extreme accuracy. A team of experts has been assembled from Department of Trauma Surgery and Department of Image Processing and Computer Graphics of University of Szeged. The goal of the team is to research and develop appropriate software and procedures capable of performing biomechanical tests and diagnosis on newly injured (human) accident victims with bone damage. The aim is to support the surgical procedure that would optimally stabilize the structural integrity of the injured bone.

The method is to acquire CT images in order to build a 3D virtual model of the bone system of the patient. The surgeon can use this model to plan the necessary operation. Having the operation plan in hand the surgeon need some support to perform the plan as precisely as it is possible. For this reason the actual position of the patient's body and the implants should be determined. The aim of this development was to give a software tool for determining these positions from CCD camera images, it is called MedNavigator.

We plan to extend our system with the ability to help the surgeon during the localization [1]. We could identify some special marked points and give real-time information, for example, where and in which angle the surgeon has to insert the implants.

Three or more cameras are installed in the operating theatre for image acquisition (see Fig.1). Our challenge to define exact three-dimensional point positions in space with the help of camera images [2]. For example, when the surgeon drills a hole into the bone according to the previously prepared plan, (s)he has to localize the starting position and the direction of the hole on the surface of the bone. That means, that the 3D virtual model of the bone, the actual bone position, and the drill should be positioned together.

To solve this problem we need to know the cameras' mapping model, the method how to determine the model, and the position from the camera images.

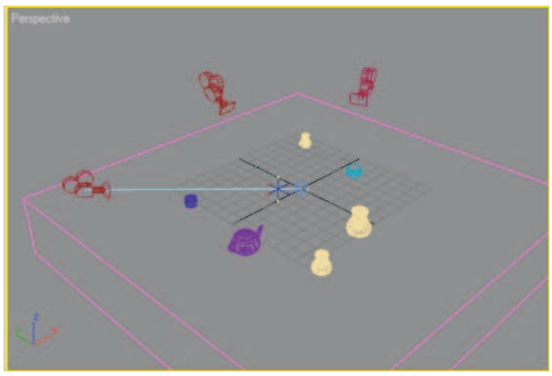

Figure 1: Used virtual space for the calibration

The camera maps the 3D space of the real world into a 2D projection. The mathematical description of this projectional mapping is necessary for the positioning. The mapping and also the mathematical description can change during the operation. For example, the camera can be moved into another place during the operation. It means that the determination of the mapping should be done quickly and easily. This procedure, called camera calibration, is the subject of this paper.

*marotikrisztina@gmail.com

2 Calibration methods

2.1 The basic types of calibration

The calibration has two basic types [5]:

1. Photogrammetric calibration.
2. Self-calibration.

Photogrammetric calibration

Camera calibration is performed by a calibration object whose geometry is known with sufficient precision. This means that the calibration object put into the camera, and then we can determine the camera's parameters with image processing method and marker points. This task requires special calibration object. For example in 3D it is usually a box.

Self-calibration

In self calibration a camera is moved in a static scene, and we can determine the camera's parameters from the excursion. In this time we don't use any calibration object. Because there are a lot of parameters to estimate, we can't obtain reliable result.

2.2 Our calibration method

The calibration is to determine the mathematical transformation or mapping as the camera maps a 3D object into a 2D image. In our system the calibration object is a 3D cross (see Fig. 2). The 3D cross is our new calibration device. Its shape makes possible that really 3D information can be used during the calibration.

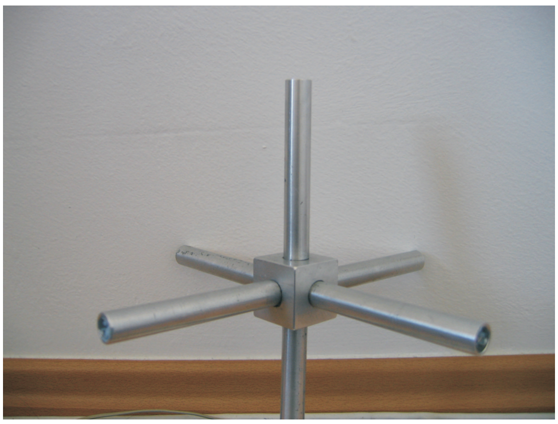

Figure 2: Our calibration object

Our calibration cross consists of 6 tubes (diameter 10 mm) forming the axes of a 3D coordinate system and there are 5 color LEDs at the ends of the tubes (the 6th end is for fixing the cross). The tubes are 10 cm long, that is, the LEDs are so far from the origin of the coordinate system represented by the cross. The flashing LEDs are the points to be identified for the calibration.

The calibration in the operating theatre must satisfy special conditions. We need a precise point definition therefore we have chosen photogrammetric calibration.

3 The mathematical description

In the following section we present the mathematical model of the camera mapping [3].

3.1 Camera calibration

The coordinate system in which we are going to determine the camera mapping is the following. The origin is the centre of the 3D calibration cross, its axes are in the same directions as the tubes of the cross. For example, the vertical tube show the axis Y.

Let M denote the matrix describing the camera mapping in the homogeneous coordinate system.

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Any 3D point

$$P = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix}$$

is transformed into the point

$$\hat{P} = \begin{pmatrix} \hat{p}_1 \\ \hat{p}_2 \\ \hat{p}_3 \\ \hat{p}_4 \end{pmatrix}$$

by this camera mapping as

$$M \cdot P = \hat{P}.$$

Our task is to determine the 12 components of matrix M, that is, we need at least 12 equations to solve such a problem. In order to find the necessary equations, let us consider the images taken from the calibration cross. Suppose that we have three cameras taking pictures from the calibration cross.

Each camera is represented with it's own mapping-matrix. Let M^i be a mapping-matrix of the ith camera, the projection of point P is $\hat{P}$ formally,

$$M \cdot P = \hat{P}.$$

Camera calibration means, that we should determine the matrix M using calibration images. Let us suppose that there are points P^j, j=1,2, ..., j, in the 3D space with known coordinates. We are going to collect images of these points with all cameras. Let the projection of point P^k by the ith camera denoted by $\hat{P^j}$, that is,

$$M^i \cdot P^j = \hat{P^{i,j}}.$$

This vector equation means three scalar equations for the first three components of $\hat{P^{i,j}}$. If we have 5 points then we have altogether 3x5=15 equations per camera, and so, 45 equations for the three cameras, which seems to be more than enough. However, due to measurement and modelling errors we cannot hope that some of the 12 from the 15 equations will be enough to determine the 12 components of any mapping matrix. For this reason instead of looking for precise solutions, we can reformulate the problem as an optimization. Formally, we want to find the solution of

$$\left\| M^i \cdot P^j - \hat{P^{i,j}} \right\|^2 \to min$$

That is, we are looking for M^i, i=1,2,3, such that the difference between the measured and the computed positions is minimal in some sense.

Powell's algorithm [4]:

To solve this we use a numerical method, Powell's algorithm, which works according to the following principle.

The inputs of the Powell's algorithm are a function and an initial direction. The algorithm minimizes the function iteratively. The result of the Powell's method are the optimal components of the matrix M^i, $i = 1, 2, 3$. We consider these matrices as the results of the calibration and the positioning procedure uses these matrices till the next calibration. As far as we know, using Powel's method for solving the calibration and positioning problems is new in the literature.

3.2 Positioning

After the calibration method where we determines the M^i matrices - we would like to determine certain points in the space. One point in an image made by a camera determine a line in a 3D space, which means that we need at least two cameras to determine a point in space. Because of noise, and errors it is better to use three cameras. As we saw in the calibration method every camera has a mapping matrix, which is calculated during the calibration method. We would like to find the point P in the space, if we know its projection. We can write down the following equation system (i=1, 2, 3)

$$M^i \cdot P = \hat{P}^i. \qquad (1)$$

Instead of solving this equation system we reformulate the problem as an optimization task, and we search a point P such that the difference of the equation system's left and right sides is minimal (least square method).

$$\left\| M^i \cdot P - \hat{P}^i \right\|^2 \to min.$$

To solve this problem we can use Powell's method again.

4 Method of calibration and positioning

In the operation theatre the first step is to make camera calibration. During calibration each of the three cameras are calibrated in the same way using the same 3D calibration cross mentioned in Section 2. The points P^j, $j = 1, 2, 3, 4, 5$, used for the calibration are just the LEDs of the cross. The projections of these points are easily recognised on the projection images if the points are swiched on and off during the acquisition. The difference of the two images (i.e., where the LED was switched on and was off) is more suitable for the detection of the ith projection of the jth LED, that is, $\hat{P^{i,j}}$.

Each camera gets the 2D coordinates of LEDs, then the program sends these coordinates to the mathematical module, which calculates the mapping matrices (see Fig.3).

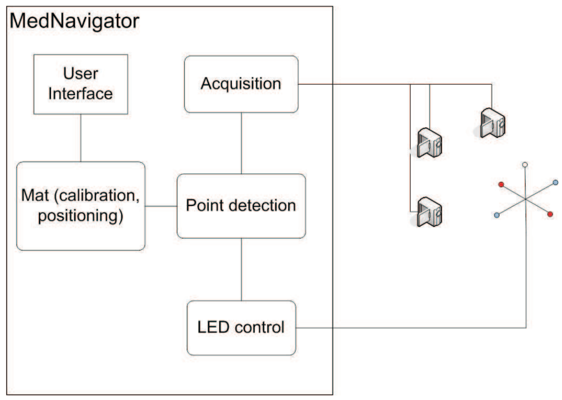

Figure 3: MedNavigator

The next step is to synchronize the camera image, the CT image and 3D model, which is made previously from CT images. During this positioning the computer switches on a marker, say P. The MedNavigator determines the positions of the projections of P in the camera images, $\hat{P}^i$, $i = 1, 2, 3$. Then using the calculated mapping matrices the program computes the 3D position P.

5 MedNavigator

MedNavigator is a collection of program modules classified functionally as follows (see Fig. 3). The Acquisition is responsible for collection of camera images. The unit LED control switches on and off the LEDs according to the control of the computer. Point detection receives the camera images and subtract the images in order to get difference images for the determination of the coordinates of the LEDs projections. This unit send the point positions for further processing to the module Mat. Mat has two basic tasks. It computes the mapping matrices of the cameras during the calibration. Also the Mat module determines the 3D points coordinates from the three camera images. Finally, the whole process can be followed through the graphical User Interface (see Fig. 4). When we start the

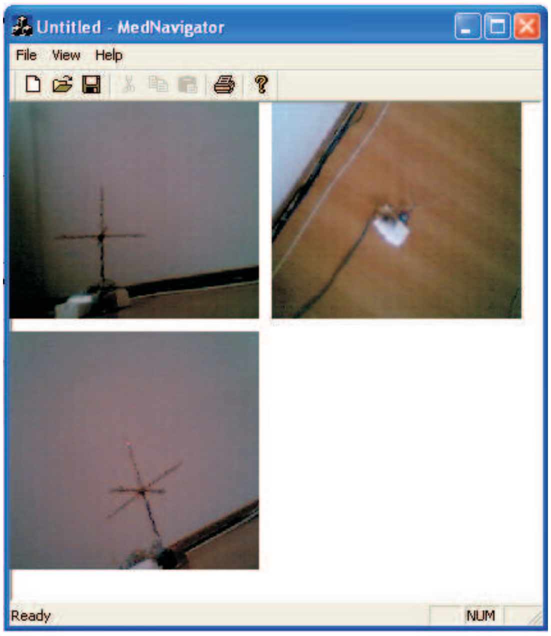

Figure 4: Window of MedNavigator showing 3 camera in use

MedNavigator we can see the three camera images. The calibration tool is connected to the computer through the parallel port, so the program can switch the LEDs. After we start the calibration the program saves the camera's images. Then it switches the LED one by one and records the images. From the initial background image it subtracts the actual image, where one of the LEDs is on, and the result is the position of the brightest point. The LED identification number, and the 2D LED coordinates are given to the Mat module. The Mat module calculates the mapping matrix.

6 Results

We tested the MedNavigator system in different ways. First, we tested the Mat module independently from the camera images. It means that the mathematical procedures got numerical input data and the output was checked knowing the exact results to be computed when the system works perfectly. For example, the camera calibration gave the mapping matrix for each camera as an output. Then we checked the equation system (see Eq. 1) using the matrix and the known point positions.

After this testing and making the necessary program modifications, we started the testing of the whole system. The camera images collected from the calibration cross were used for this validation. From these images we determined the LED positions. Knowing the LED positions exactly we could compare the computed coordinates with the real ones.

The measured and the real coordinates of the LEDs and the Euclidean distance between them (row D) are in Table 1.

The table shows that the differences are about half a centimeter. Such an error is too big for our aim, so we started to look for the reasons of the errors and to correct the program. Finding and solving such problems like inexact positioning due to wrong calibration object, synchronisation and high intensity points due to mirroring. We could reduce this error as it is seen from Tables 2 and 3. Table 2 shows the best result we have while the Table 3 is the worst result. Table 4 shows the summary of 100 tests results. The first column indicates the number of test with the error of the second column. We emphasise that 81% of own tests performed under 0.3 cm.

	1.LED	2.LED	3.LED	4.LED	5.LED
X^*	10	-10	0	0	0
Y^*	0	0	0	10	-10
Z^*	0	0	10	0	0
X	9.58	-10.54	0.00	0.41	0.53
Y	-0.10	-0.24	0.00	10.10	0.20
Z	-0.20	-0.18	10.00	0.20	0.23
D	0.48	0.65	-0.00	0.48	0.60

Table 1: Measured (X^*, Y^*, Z^*) and the real (X, Y, Z) coordinates in cm and D is the Euclidean distance

Our computation platform was Microsoft Windows XP. We used the compiler Microsoft Visual Studio .Net 2003. The system needs 4-5 seconds for collecting the images, 1-2 seconds for point detection, and less than 1 second for computation. The navigation system needs 150 KB memory, and minimum 100 Mb working space.

	1.LED	2.LED	3.LED	4.LED	5.LED
X^*	-11,2	11	0,5	0,3	0,5
Y^*	-1,8	-1,8	10	-1,2	-1,3
Z^*	0	0	0	-10,7	10,5
X	-11,10	11,25	0,51	0,19	0,34
Y	-1,72	-1,80	10,00	-1,28	-1,39
Z	-0,010	0,039	-0,00	-10,68	10,37
D	0,12	0,26	0,01	0,13	0,22

Table 2: Best measured (X^*, Y^*, Z^*) and the real (X, Y, Z) coordinates in cm and D is the Euclidean distance

	1.LED	2.LED	3.LED	4.LED	5.LED
X^*	11,2	-11	0,5	0,3	0,5
Y^*	-1,8	-1,8	10	-1,2	-1,3
Z^*	0	0	0	-10,7	10,5
X	-10,98	11,12	0,52	0,19	0,08
Y	-1,68	-1,77	10,01	-1,32	-1,35
Z	0,13	0,09	0,01	-10,78	10,37
D	0,28	0,15	0,02	0,18	0,43

Table 3: Worst measured (X^*, Y^*, Z^*) and the real (X, Y, Z) coordinates and D is the Euclidean distance

Number of result	Error-Distance
3	0,4-0,5
16	0,3-0,4
52	0,2-0,3
10	0,1-0,2
19	0,0-0,1

Table 4: Result of 100 tests

7 Conclusion

We found that the best positioning error in this system is about 0.3 cm. In order to make further improvements we think that we should change our cameras. In the present system the resolution of the cameras is 320*240. The same procedure can be applied if the cameras have better resolution. We expect that if the cameras are giving 640*480 resolution images then then error can be half of the present value. The system has an own web page:
http://www.inf.u-szeged.hu/ medsys/ .

8 Acknowledgements

The author thanks to Dr Attila Kuba , Dr Endre Varga, György Bekes, Balázs Erdőhelyi, Csongor Halmai and Krisztián Ollé for their help during the experiments and writing the paper. This work was supported by the grant OTKA T37840 of the Hungarian Research Foundation (OTKA).

References

[1] General Electric Company. *Image Guided Surgery*. http://www.gehealthcare.com/rad/savi/nav/igs.html, 2004.

[2] Ph.D. Gabor Fichtinger. *Surgical Navigation, Registration, and Tracking*. 2004.

[3] Thorpe C.E. Kanade T. Gremban, K.D. *Geometric camera calibration using systems of linear equations*. Robotics and Automation, 1988 IEEE International Conference, 1988.

[4] W. T. Vetterling B. P. Falnnery W. H. Press, S. A. Teukolsky. *Numerical Recipes in C*. Cambridge University Press, 1992.

[5] Z. Zhang. *A flexible new technique for camera calibration*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.