A CALIBRATION PROGRAM FOR AUTONOMOUS VIDEO BASED ROBOT NAVIGATION SYSTEMS

Stefan Florczyk

Institute for computer science, Munich University of Technology, Boltzmannstr. 3, D- 85748 Garching, e-mail: *florczyk@in.tum.de*

Abstract. A program for camera calibration was implemented for a robot vision project. The aim of this project is the development of software to generate a map with an autonomous mobile robot. The map generation must be performed only video based and shall be the base for tasks which must be fulfilled by an office messenger or a watchman. The field for service robots requires especially cheap, fast, and simply portable software. Calibration program SICAST (simple calibration strategy) in this work meets these requirements.

1. Introduction

We are working at a video based indoor navigation system for autonomous robots. The navigation shall be based on a map [1] which must be generated autonomously by a robot. This map shall be the base for tasks, which an office messenger or a watchman has to do. Several researchers [2] are currently working in generating such video based maps, but at this time a reliable application doesn't exist.

To generate such a map a robot must be able to determine the three-dimensional world co-ordinates $(x^j, y^j, z^j)_W$ with $j = 1, 2, \ldots, n$ of n points in the world co-ordinate system W. These points shall belong to relevant objects for navigation and collision avoiding. Therefore, it is necessary to calibrate a camera, which is mounted on our Pioneer 1 robot. Indoor exploration requires the use of wide-angle-lenses. This aggravates the calibration procedure, because such lenses show a fisheye effect with distortions. The expected practical fields of autonomous service robots yield further requirements:

1. Service robots are only profitable if their production costs are rather low.
2. A service robot must work in real-time.
3. A service robot must be able to run on different operating systems.

So we are enforced to develop very cheap software. The issues 2 and 3 correlate with issue 1. A robot which doesn't fulfill its tasks in real-time has too low productivity. Therefore, very fast algorithms are important. Issue 3 safes also costs. A service robot, which runs on different operating systems, can be sold for lower costs because of its expected larger sale in comparison to a robot, which only runs on one operating system. In this work a program for camera calibration is depicted which meets these requirements. Several calibration approaches can be found, but most implementations are working with license software like MAPLE or MATLAB or the calibration procedure is very time-consuming. In chapter 2 there are three calibration approaches explained. Chapter 3 discusses important algorithms, which are used by SICAST. The eligibility of the calibration program has been tested in chapter 4. In last chapter it is given a conclusion and an outlook.

2. Camera calibration for indoor exploration with robots

Camera calibration is an important prerequisite for successful video based indoor room exploration. It allows us to determine the parameters, which describe the projection of three-dimensional world co-ordinates $(x^j, y^j, z^j)_W$ of a point j in the world co-ordinate system W to its two-dimensional image co-ordinates $(u^j, v^j)_I$ in the image co-ordinate system I. Let x_W^j and u_I^j be the depth distance, y_W^j and v_I^j be the lateral distance, and z_W^j the height distance to an origin belonging to W or I. So it is possible to generate a reliable map by using examined three-dimensional world co-coordinates. This map can then be used for service tasks like an office messenger performed by a robot. Many approaches for calibration exist. Commercial image processing program HALCON contains an operator [3] to determine internal and external parameters of a camera. The approach needs at least 10 to 20 images of a calibration object, which can be a planar calibration table. An object must be taken from different positions and orientations and an object should fill at least a quarter of an image. Using the HALCON operator requires also some knowledge about technical data of a camera, which are used as initial value. This is a nominal focal length f, a horizontal size s_x and vertical size s_y of a cell on a CCD (charge coupled device) sensor. These initial values must be given in real word units (meters). Co-ordinates $(u^c, v^c)_I$ representing a center point c of an image are also necessary. An initial value can be determined by bisection of image width and image height. These values must be determined in unit pixel. Calibration method of Roger Tsai is widespread in industrial applications and has been described in [4]. It is assumed that a principal point $p\,(u^p, v^p)_I$, which represents a point of intersection of a camera's optical axis and an image plane, is known. Further necessary data are scale factors d_x and d_y representing a horizontal and vertical size of a pixel in a two-dimensional image and an aspect distortion factor τ to capture distortion regarding ratio of image width and image height. The determination of focal length f and radial distortion κ was performed. Strategy is similar to HALCON's calibration operator. A sheet of paper showing a calibration table was affixed onto a vertical plate. This vertical plate could be displaced, because it was mounted on a rail. The construction makes it possible to modify a distance between a calibration object and a camera by a very precise known distance. Moving of an object happens in 10 millimeters steps. Approach of Faugeras elucidated in [5] performs calibration for a pinhole camera model. There must be no initial knowledge of a camera's technical data. After calibration the technical data are delivered in a calibration matrix K, which can be derived from a projective matrix P. A projective matrix P performs a projection of three-dimensional world co-ordinates $(x^j, y^j, z^j)_W$ onto two-dimensional image co-ordinates $(u^j, v^j)_I$. First step is the determination of P which happens with a linear equation system. This linear system models the correspondence between $(x^j, y^j, z^j)_W$ and $(u^j, v^j)_I$. It is necessary to take at least 6 points. Often it is required to take much more points. Determination of belonging three-dimensional world co-ordinates $(x^j, y^j, z^j)_W$ and corresponding two-dimensional image co-ordinates $(u^j, v^j)_I$ can happen with a calibration object from which one image was taken. Points may not be degenerated that means that different points must have different values for its x_W^j, y_W^j, and z_W^j co-ordinates. Every correspondence between a three-dimensional world point with $(x^j, y^j, z^j)_W = (x^j, y^j, z^j)$ and a two-dimensional image point $(u^j, v^j)_I = (u^j, v^j)$ with $j = 1, 2, \ldots, n$, whereby n is the number of points for which correspondences are modeled, delivers two equations:

$$\begin{pmatrix} x^1 & y^1 & z^1 & 1 & 0 & 0 & 0 & 0 & -u^1x^1 & -u^1y^1 & -u^1z^1 & -u^1 \\ 0 & 0 & 0 & 0 & x^1 & y^1 & z^1 & 1 & -v^1x^1 & -v^1y^1 & -v^1z^1 & -v^1 \\ & & & & & & \vdots & & & & & \end{pmatrix} \begin{pmatrix} p_{11} \\ p_{12} \\ \vdots \\ p_{34} \end{pmatrix} = 0. \tag{1}$$

The equation system has $2n$ rows. If more than 6 points are used, an over-determined equation system must be solved. This can happen with an algorithm using the statistic method of least squares. Determined values for unknown parameters establish $\boldsymbol{P}$:

$$\begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{pmatrix} x_W \\ y_W \\ z_W \\ 1 \end{pmatrix} = \begin{pmatrix} d\hat{u}_i^j \\ d\hat{v}_i^j \\ d \end{pmatrix}. \tag{2}$$

d is a scaling factor. $\hat{u}_i^j$ and $\hat{v}_i^j$ are two-dimensional co-ordinates, which are examined by re-projection and therefore denoted with the roof. The calibration matrix $\boldsymbol{K}$ can be derived from $\boldsymbol{P}$. It is also possible to determine a rotation matrix $\boldsymbol{R}$ and a translation vector $\mathbf{t}$ which allow the conversion between the world co-ordinate system W and the camera co-ordinate system C:

$$\boldsymbol{P} = (\boldsymbol{KR} \mid -\boldsymbol{KR}\mathbf{t}) = (\hat{\boldsymbol{P}} \mid \mathbf{p}_4). \tag{3}$$

$\hat{\boldsymbol{P}}$ is the product of $\boldsymbol{K} \cdot \boldsymbol{R}$ and a 3×3 sub matrix of $\boldsymbol{P}$. $\mathbf{p}_4$ is the rightmost column of $\boldsymbol{P}$. The splitting of $\boldsymbol{P}$ is noted with the delimiter. If $\hat{\boldsymbol{P}}$ is known, the translation vector $\mathbf{t}$ can be processed:

$$\mathbf{t} = -\hat{\boldsymbol{P}}^{-1}\mathbf{p}_4. \tag{4}$$

Using the fact that $\boldsymbol{K}$ is upper triangular and $\boldsymbol{R}$ is orthogonal makes it possible to use a QR-decomposition [6] which decomposes $\hat{\boldsymbol{P}}$ into two matrices which such features.

3. SICAST – a calibration program for indoor exploration

In this work the approach from Faugeras was adopted, because it provides some advantages for our exploration project. A very simple camera with a wide-angle-lens is mounted on our Pioneer 1 robot and transmits images to a framegrabber, which is mounted in a static computer. We have no information about the technical data of the camera. So it is not possible to use neither HALCON's operator nor Roger Tsai's approach. A second advantage of Faugeras's approach is the simple strategy. Especially it is not necessary to take images from many different positions and orientations. Faugeras and Mourrain [7] used MAPLE to perform their mathematical calculations. In contrast the program SICAST uses only the standard C++ libraries and some algorithms for QR-decomposition and singular value decomposition (SVD) which can be found in [6] as well as some algorithms for matrix calculations described in [8] to fulfill some requirements necessary for our project:

1. The source code is known.
2. The program is simply portable.
3. The relinquishment of commercial libraries safes costs.

Consistent with issue three there was also used an own three-dimensional calibration object. Only a framegrabber and a C++ compiler must be bought to perform the camera calibration. Calibration object consists of two tins of same size, which have been set one upon another. The tins have been partially wrapped around with paper. So it was possible to draw marks on the paper in different heights. Non-degenerative points can now be gained if the object's position is changed in every image. Figure 1 shows the calibration object:

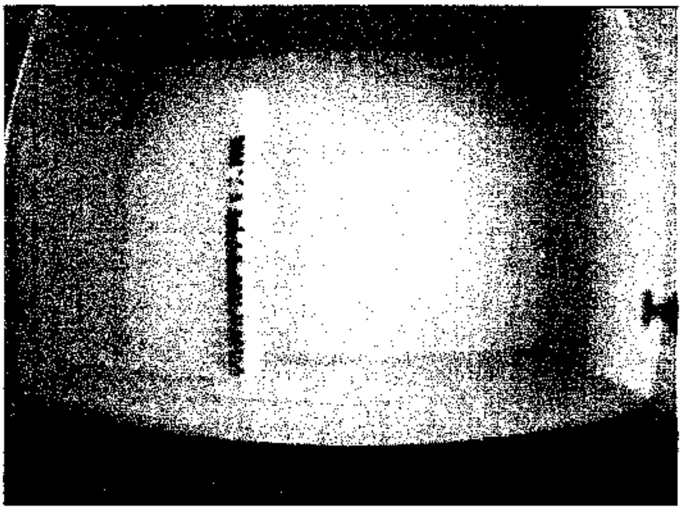

Figure 1: The used three-dimensional calibration object.

The positioning of the object happened within an office cupboard to ease the determination of x_W^j and y_W^j co-ordinates. Before a new image was taken, there was marked a dot onto a sheet of paper with standardized squares. This dot represented the x_W^j and y_W^j co-ordinates. Precise positioning of the calibration object was then very simple. Note also that the calibration process is aggravated because of the strong distortions, which can be observed in the image.

4. Experiments

Thirty points have been recorded with the three-dimensional calibration object shown in figure 1. It was used a camera with a wide-angle-lens. Such lenses are typical for indoor exploration. Strong distortions aggravate the calibration. The camera was mounted on a Pioneer 1 robot. The selected points were not degenerated to fulfill the requirements, which means that no value for x_W^j, y_W^j, z_W^j with $j = 1, 2, \ldots, n = 30$ has been taken twice. Every point was taken in a new image with a different position and height mark. The robot and the mounted camera haven't been moved during the image takings. Then it were determined the three-dimensional world co-ordinates and two-dimensional image co-ordinates for every point. The solution of the resulting equation system (1) has then been examined with SICAST. The quality of calibration results was evaluated by re-projection of measured three-dimensional points $(x^j, y^j, z^j)_W$ with the gained projection matrix using equation (2). The calculated terms $d\hat{u}_I^j$ and $d\hat{v}_I^j$ have been divided by d. The computed two-dimensional image co-ordinates $(\hat{u}^j, \hat{v}^j)_I$ with $j = 1, 2, \ldots, n$, whereby n is the number of actually considered points, could then be compared with the empirically measured two-dimensional image co-ordinates $(u^j, v^j)_I$ using the following equations which process the mean deviations u_{mean} and v_{mean} in u and v direction:

$$u_{mean} = \frac{1}{n} \sum_{j=1}^{n} \left| u_I^j - \hat{u}_I^j \right| \tag{5}$$

$$v_{mean} = \frac{1}{n}\sum_{j=1}^{n}\left|v_i^j - \hat{v}_i^j\right|. \quad (6)$$

For $n = 30$ the value for u_{mean} was 33.1 pixel and for v_{mean} 40.9 pixel, but the processing was hardly damaged by fife outliers who could be simply detected with reprojection. After elimination of these clear outliers ($n = 25$) u_{mean} got the value 18.8 pixel and v_{mean} the value 20.6 pixel.

5. Conclusion and outlook

In this work three approaches for camera calibration were examined for our robot vision project. Two approaches need initial values and the calibration procedure is very time-consuming. The third declared approach didn't show these drawbacks, but its implementation was not appropriate for our robot vision project, because it uses the commercial library MAPLE. Our robot navigation system must be simply portable and therefore we are enforced to use only standard libraries. Additionally we are forced to hold the development costs as low as possible and we are interested in the source code. These requirements make it necessary to develop an own calibration program adopting the third explained approach. An experiment was performed. The experiment showed that the calibration was difficult because of the used wide-angle-lens. Additionally the first processing with all measured points was damaged by outliers. Elimination of these outliers improved the quality of the projective matrix.

References

[1] Thrun, Sebastian/ Burgard Wolfram/ Fox Dieter: A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots, Machine Learning, Vol. 31, pp. 29-53, 1998.

[2] Florczyk S, 2001, "A Redundant Program to Handle Inhomogeneous Illumination and Changing Camera Positions in a Robot Vision Scenario", in *Pattern Recognition: Methods and Applications, Proc. VI Iber-American Symp. Pattern Recognition* (SIARP 2001), IAPR-TC3, Florianypolis, Brazil, 100-105.

[3] MVTec, 2000, *HALCON / C++: Reference Manual*, CD-ROM, MVTec Software GmbH, Munich.

[4] Shapiro L G and Stockman G C, 2001, *Computer Vision*, Prentice Hall, Upper Saddle River, N. J.

[5] Sonka M, Hlavac V, and Boyle R, 1999, *Image Processing, Analysis, and Machine Vision*, 2. ed., PWS Publishing, Pacific Grove, Calif. et al.

[6] Press W H et al., 1992, *Numerical Recipes in C: The Art of Scientific Computing*, 2. ed., Cambridge Univ. Press, Cambridge.

[7] Faugeras O and Mourrain B, 1995, *On the Geometry and Algebra of the Point and Line Correspondences between N Images*, Technical Report 2665, Institut National de Recherche en Informatique et en Automatique, Sophia-Antipolis, France.

[8] Janiszczak I, Knurr R, and Michler G O, 1992, *Lineare Algebra für Wirtschaftsinformatiker: Ein Algorithmen-Orientiertes Lehrbuch mit Lernsoftware*, Vieweg, Braunschweig, Wiesbaden, Germany (in German).