A Novel Corner Detection Algorithm for Camera Calibration and Calibration Facilities

THEODORE PACHIDIS*, JOHN LYGOURAS**, VASILIOS PETRIDIS ***
*Department of Electrical and Computer Engineering
Democritus University of Thrace
67100 Xanthi
GREECE
pated@mail.otenet.gr
**Department of Electrical and Computer Engineering
Democritus University of Thrace
67100 Xanthi
GREECE
ilygour@ee.duth.gr
***Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
Thessaloniki
GREECE
petridis@vergina.eng.auth.gr

Abstract: - A novel corner detection algorithm is presented which can be used to camera calibration methods where square corners are used as control points. Corners are detected with sub-pixel accuracy, using a segmentation method for separation of each square, based on seeds. These are pixels with a predefined color or gray value. An 11x11 proper developed template, including pixels of the predefined color or gray value, convoluted with the corresponding square gives pixels related with a corner. The mean value of this cluster of pixels provides with sub-pixel accuracy the co-ordinates of the specified corner. Corners co-ordinates are calculated with the specified sequence of the camera calibration method. Corners are found even in cases where the square slope is big or the barrel phenomenon distorts too much the image. The software interface was made in visual C++. Some of its features are the possibility to change the scanning area making the algorithm faster or more reliable, saving facilities for the converted binary image and for the final corners file, model file creation. This program is part of a software application where images can be easily captured using a camera mounted on the end effector of PUMA 761 robotic manipulator and calibration is made through Z. Zhang method using a novel easy selectable data files method which is provided with the same program.

Key-Words: - corner detection, camera calibration, sub-pixel accuracy, interface, model generation, facilities.

1 Introduction

The calibration of a camera is generally a necessary process when it is to be used as a sensor for measurement. Then problems with axial distortion can be eliminated and the accurate calculation of the active focal length and of the center of the image is possible. The most calibration methods use 3D or 2D patterns of predefine shapes (i.e. squares, circles, chessboard, e.t.c.) In these patterns the control points can be the centers of gravity of circles or squares, intersections of lines in rectangular or triangular form, corners of squares or chessboards [1]. In patterns with shapes in a rectangular form (squares or chessboard pattern) the accurate calculation of the co-ordinates of corners is required. Many corner detection algorithms have been developed recently where corners are estimated and final corners co-ordinates are found. Most of them are used to gray scale images. These algorithms can be classified in two main categories [2]: template-based corner detectors and geometry-based detectors. In template-based corner detectors, a set of corner templates is developed and the similarity between the templates and all the sub-windows of a gray level image is determined. This kind of corner detectors is not widely used because of its high algorithm computational cost. Geometry-based corner detector relies on measuring differential geometry features of corners. Some of them are widely used in all kinds of applications.

A technique for detecting and localizing corners of planar curves is proposed from A. Rattarangsi and

R. Chin in [3]. The technique is based on Gaussian scale space. Deriche and Giraudon in [4] dealt with the development of a computational approach to corners and vertices. A corner model is considered and its behavior is analytically studied. Two corner detectors are presented from Cooper et al. [5], the first using dissimilarity along the contour direction to detect curves in the image contour and the second estimating image curvature along the contour direction. Tabbone [6] described a fast corner detection algorithm making use of the Laplacian of Gaussian operator. Brand and Mohr [7] in their paper show how image points can be extracted accurately. Their search was restricted to specific points identified by corners. F. Mokhtarian and R. Suomela in [8] presented a novel method for image corner detection based on the curvature scale-space (CSS) representation. Smith et al. in [9] tackled the problem of obtaining a good initial set of corner matches between two images without resorting to any constraints from motion or structure models. Ruson and Tomasi used both, a region model and an edge model to perform corner detection on color images whose regions contain texture [10]. Li and Chen used fuzzy logic to solve the problem of corner detection on planar curves [11]. In [12], S. Ando applied gradient covariance for edge/corner detection. Shen and Wang in [13] proposed a local edge detection algorithm that can be used well in corner detectors. Finally Alvarez, Cuenca and Mazorra, [14] studied the application of the Affine Morphological Scale Space (AMSS) to corner detection with subpixel accuracy. Their technique of corner detection was applied to calibrate a camera system using a calibration object. Many other researchers have developed a variety of algorithms and methods for corners detection.

Our algorithm is a specific algorithm and can be used in combination with Z. Zhang calibration method. It can be also used as a general corner detection algorithm for gray and color images with small changes. It calculates with sub-pixel accuracy the corners of squares of a pattern and saves them with the specified order to a data file, ready to be used for calibration. Some features and advantages of this algorithm are the following:

a. Squares to the pattern are segmented using small scanning area.
b. The concept of seeds is used.
c. A proper sub-image (template) is convoluted with the captured or opened image for corner detection.
d. It can find corners in images with big slope of squares.
e. When the procedure is finished the converted image is presented with the different intensity values and crosses to each corner with pixel accuracy so that the appearance of the final result is easy controlled.
f. It is fast enouth and reliable.
g. It can be used, combined with the developed software, for camera calibration in short time.
h. The model of the pattern can be automatically calculated and be saved to a predefined file.

The paper is organized as follows. In Section 2, the basic concepts, the corner detection algorithm and the software interface are presented. In Section 3, the experimental results are presented. Finally, in Section 4, the conclusions of this paper are given.

2 Basic Concepts, Corner Detection Algorithm and Software Interface

2.1 Basic Concepts

For camera calibration the input data must be as accurate as possible. It is also necessary these data to be easily, reliably and in a short time exported of an image so that the calibration procedure to be repeated as often as it needs. The proposed corner detection algorithm, can be used mainly with Tsai [15] and Z. Zhang [16] camera calibration method for squares corners extraction.

A corner detection algorithm should satisfy some important criteria:

1. All the true corners should be detected with the specified order by the algorithm.
2. No false corners should be detected.
3. Corner points should be calculated with sub-pixel accuracy.
4. The algorithm should be robust with respect to noise.
5. It should be able to accurately find corner points with big slope of the squares.
6. It should be accurately find corner points in images with large values of distortion where barrel phenomenon is presented.

Two basic concepts are used. The first one concern the method of separating each square of the binary or gray scale or color image. Here a binary image is selected for the analysis but the method can be easily extended to gray scale and color images. For this reason the concept of seed was used. That is a pixel with a unique specified color or gray scale value. In patterns, as shapes can be considered as separated entities, in each shape (i.e. a square) a seed is planted. Then, this seed can be propagated only to this entity. In this way, the segmentation of

an image is possible and if it is desired with a specified order of the created segments. The last idea is implemented to the proposed algorithm. The segmentation order is specified using as features for the estimation of the new seed the center and the slope of the previous segmented square. Two criteria for the selection of a seed are used. The first is its distance from a specified initial point. As second criterion is taken the following: the candidate pixel must be the center pixel of a 5X5 template with cross shape (Fig. 3(b)). The last criterion is used to make the pixel selection insensitive to noise.

The second concept concerns the used method for corner detection. In introduction was mentioned that templates-based algorithms are slower than geometry-based algorithms. Here, only two 11X11 templates (Fig. 1, Fig. 2) are used. In these templates a small number of pixels convoluted with the image pixels. The scanning area is also limited to a region around the current square with the seed. This way the algorithm is quite fast. The orientation of the templates depends on the orientation of the searching corner. In Fig. 1 and 2 "S" represents the substrate color, "C" the square color and "X" are no interested color. In the first template (Fig. 1) the center pixel is type "C" and in the second template (Fig. 2) the center pixel is type "S". As it is easily realized from these figures the templates are "X" shape where the one leg is colored with the square color and the other three legs are colored with the substrate color. The detecting pixel is the center pixel of the templates. Rotating these two basic templates 90^{o} each time and then applying them to the image the detection of all corner pixels is possible. The interesting corner point is the peak where is formed around corner area from the change of the substrate color to square color and vice-versa. This peak may be the mean value of two pixels in a well-formed square in a binary image or the mean value of many detected pixels, creating a cluster of pixels around the corner, using the above templates in a binary, gray scale or color image. Using these slightly different types of templates the inner and the outer side pixels of the corner are simultaneously detected.

2.2 Corner Detection Algorithm

The proposed corner detection is the following:

a. Scan the image, scanning first from the bottom left corner.
b. Find the first square pixel (i.e. a black pixel if the image is binary). This pixel is the center of a 9X9 template where only some pixels are used. These pixels form a cross shape (fig. 3(a)). The reason of the selection of the above template is again to be insensitive to noise.

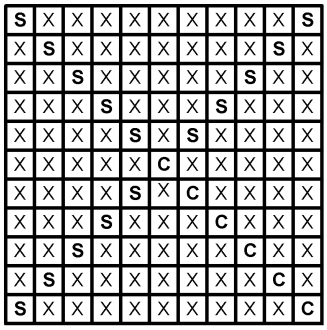


Fig. 1 Template 11 X 11 with square color "C" center pixel.

c. Check if this pixel belongs to the first or last shape, scanning in the same horizontal line. If it doesn't belong then a pixel is selected of the detected pixels in this line, using as criterion its location.
d. Create a seed changing the color of this pixel to the automatically specified new color or gray scale value.

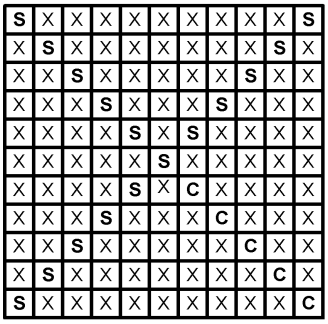


Fig. 2 Template 11 X 11 with substrate color "S" center pixel.

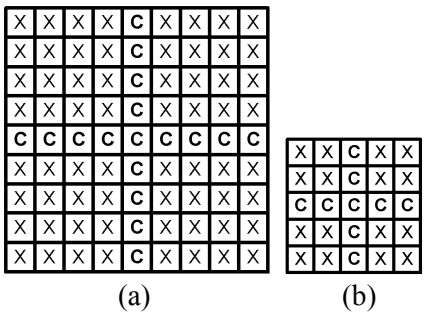


Fig. 3 (a) 9 X 9 template for detection of the square pixels during the segmentation procedure (b) 5 X 5 template for detection of pixels during the scanning of a line.

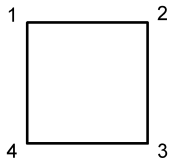


Fig. 4 Angles order.

e. Propagate this seed to the whole shape (square) scanning in a small area, which is defined considering that the pixel coordinates are the center of a rectangular area and the limits of it are predefined.
f. From this first square, find following the specified order (Fig 4), all possible corner pixels. For each corner, store all possible corner pixels to a sub-matrix.
g. Find the mean value of these pixels. This value is the final corner value with sub-pixel accuracy. Store this value to a corner matrix.
h. Calculate the center of the square as the mean value of the four previous calculated corner values.
i. Calculate the slope of the square. The slope is calculated of the pixel pairs (1,2) and (3,4) and taking the mean value of them.
j. Scan the image along the line passing through the previous defined center or through a pixel of the square where its position is calculated from the center and the slope of this square. Store all pixels with the predefined square color.
k. Select of the previous pixels, the nearest to the initial pixel. This pixel is the center of a 5X5 template (Fig. 3(b)) with cross shape also (noise insensitivity).
l. Create a new seed changing the color of it with an automatically specified new color or gray scale value.
m. Check if the squares centers were horizontally found. If they were found, continue. Else repeat steps e-l.
n. Using now as origin the center of the first square horizontally detected and by means of slope, find all pixels, vertically scanning along a line. These pixels are again the center of a 5X5 template with cross shape, thus, avoiding pixels not belonging to a square.
o. Select the nearest pixel to the previous initial pixel as seed.
p. Check if all squares centers were found. If they were found, continue. Else repeat steps e-o.
q. Scanning the whole image, clear the useless pixels (giving them the substrate color, i.e. white)
r. Put with pixel accuracy crosses to each corner point and display the reformed image.
s. Save the image with a predefined file name. (i.e. data01.txt).

2.3 Software Interface Brief Description

A proper program was developed in Visual C++ for camera calibration using the Z. Zhang method. This program is part of an integrated software application where the communication via ALTER port and the control of a PUMA 761 robotic manipulator is possible using as sources, data of image processing by means of the constructed pseudo stereo vision system (PSVS), data files or directly using a mouse. The user interface, referred to corner detection and camera calibration, consists of three parts. The first one (Fig. 5) provides all the facilities for image capturing, saving, corner detection and saving of the data with a specified file name. The model file can also be created using as image the pattern design. Some properties of the calibration procedure (i.e. the names of the data files), as also of the corner detection algorithm can be regulated selecting the "Calibration Properties" function. Using the other buttons the camera calibration is possible either step by step or automatically.

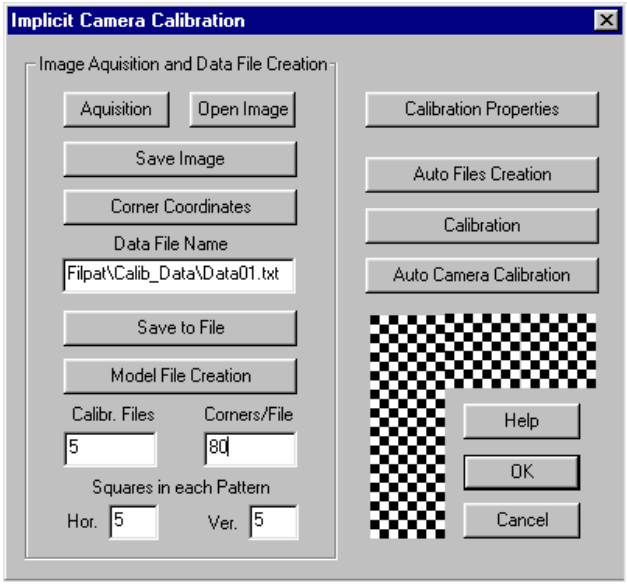


Fig. 5 Corner detection and camera calibration software interface.

The automatic calibration is supported by the second interface (Fig. 6) where using the loaded data of the manipulator position and orientation, the system can capture sixteen (maximum) images from different locations, calculate the corner data files, calibrate the camera and display the results. The manipulator proper locations could initially be

generated using the second interface and then to be saved to a text file.
The third part of this interface is used as display for images, where using a mouse, the location and the intensity of each pixel is appeared. This is used to easy check the whole procedure. In Fig. 7, a pattern of squares in its final form is illustrated after the implementation of the corner detection algorithm. Crosses with pixel accuracy show the corner locations in each rectangle.

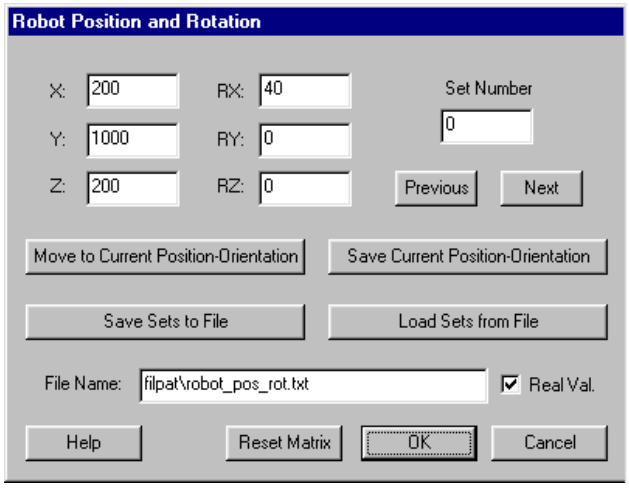


Fig. 6 Software interface for the movement of the manipulator to new or predefined location.

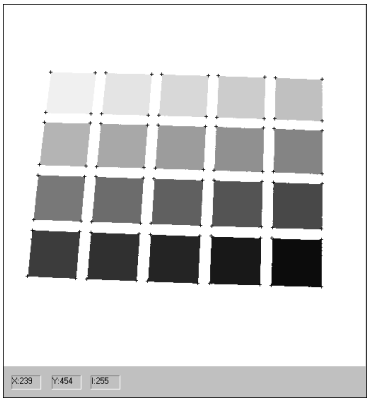
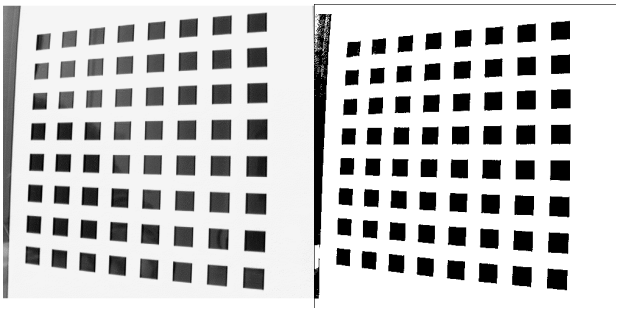


Fig. 7 Interface where an image is displayed to easy check the final result.

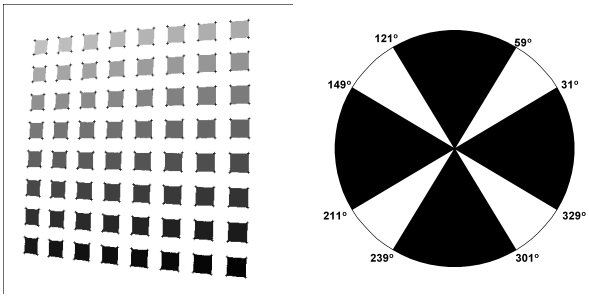
3 Experimental Results

In order to check the proposed algorithm a series of experiments were made. To check how fast the corner detection algorithm was working a timer was introduced to the program to measure the algorithm running time. Many trials were made using two different patterns and many different views of these patterns. The measured time was always between 0.26 and 0.50 sec. This means that the camera calibration procedure considered and the movement of the manipulator, using four to five images can take less than one minute when the system works automatically.



(a) (b)

Fig. 8 (a) Iinitial image (b) Binary image



(a) (b)

Fig. 9 (a) Final image after the implementation of the corner detection algorithm, (b) Square slopes range.

To check the proposed algorithm, an image of Z. Zhang method and the corresponding data file with the corner locations are used. These data are found to the web site of Microsoft. The target was to compare the results of the proposed algorithm with the results calculated with the Z. Zhang method corner detection algorithm. The selected image was converted to bitmap and the size of substrate was changed to 512x512 pixels without any change to the square size. This was made because the used image size was different from that of the initial image. In Fig. 8(a), (b) the converted original image and the corresponding binary are illustrated. In Fig. 9(a) the resulting image is presented. Crosses designed with pixel accuracy are distinguished to the corners of each square. The ordered segmentation is also presented in the same image as different intensity values. The resulting corner locations of the proposed algorithm were similar with the corner locations of Z. Zhang data file.

Finally a third experiment was made with purpose to check the influence of the squares slope in a pattern to the reliability and the accuracy of the corner detection algorithm. For this reason many 512X512 images with the same pattern of squares (four squares) but with different slopes, in steps of 5^o or 1^o were created. Implementing the proposed algorithm to these images, the angles of squares slope where the algorithm can reliably and accurately find corners were found and are illustrated in Fig. 9(b) with black color.

4 Conclusion

In this paper a novel corner detection algorithm was presented. This algorithm and the related software were developed for easy, fast and reliable calibration of a camera. This camera was mounted to the end effector of a PUMA 761 robotic manipulator. Many facilities are offered from the related software. A three parts user interface is offered where a user could calibrate a camera step by step (capturing or opening an image, finding corners from different images, saving images and data files and finally calibrating the camera) or automatically using predefined manipulator locations to capture different views of a pattern. The corner detection algorithm is quite fast, reliable and corner points are calculated with sub-pixel accuracy. The corners points are saved with a predefined order. The model file is also generated using as image the pattern design. It can be used with any calibration algorithm where the control points are corners of squares.

References:

[1] H. Bakstein, A Complete DLT-based Camera Calibration with a Virtual 3D Calibration Object, Faculty of Mathematics and Physics (Diploma Thesis), 1999.

[2] F. Shen and H. Wang, Real Time Gray Level Corner Detector, 6th International Conference on Control, Automation, Robotics and Vision (ICARCV2000), 2000.

[3] A. Rattarangsi and R. T. Chin, Scale-Based Detection of Corners of Planar Curves, IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 14, NO. 4, 1992, pp. 430-449.

[4] R. Deriche and G. Giraudon, A Computational Approach for Corner and Vertex Detection, International Journal of Computer Vision, VOL. 10, NO. 2, 1993, pp.101-124.

[5] J. Cooper, S. Venkatech and L. Kitchen, Early Jump-Out Corner Detectors, IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 15, NO. 8, 1993, pp. 823-828.

[6] S. Tabbone, Corner Detection Using Laplacian of Gaussian Operator, Scandinavian Conference on Image Analysis, 1993.

[7] P. Brand and R. Mohr, Accuracy in Image Measure, In Videometrics III, Proc. SPIE 2350, 1994.

[8] F. Mokhtarian and R. Suomela, Robust Image Corner Detection Through Curvature Scale Space, IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 20, NO. 12, 1998, pp. 1376-1381.

[9] P. Smith, D. Sinclair, R. Cipolla and K. Wood, Effective Corner Matching, 9th British Machine Vision Conference (BMVC'98), 1998.

[10] M. A. Ruzon and C. Tomasi, Corner Detection in Textured Color Images, IEEE 17th International Conference on Computer Vision, 1999.

[11] L. Li and W. Chen, Corner Detection and Interpretation on Planar Curves Using Fuzzy Reasoning, IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 21, NO. 11, 1999, pp.1204-1210.

[12] S. Ando, Image Field Categorization and Edge/Corner Detection from Gradient Covariance, IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 22, NO. 2, 2000, pp.179-190.

[13] F. Shen and H. Wang, A Local Edge Detector used for Finding Corners, Third International Conference on Information, Communications and Signal Processing (ICICS' 2001), 2001.

[14] L. Alvarez, C. Cuenca and L. Mazorra, Morphological Corner Detection. Application to Camera Calibration, IASTED International Conference on Signal Processing, Pattern Recognition & Applications, 2001.

[15] R. Y. 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. RA-3, NO. 4, 1987, pp. 323-345.

[16] Z. Zhang, A Flexible New Technique for Camera Calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 22, NO. 11, 2000, pp.1330-1334.