Image Processing and Recognition System for a Robot Arm Control

St.Gh. Pentiuc, Radu Vatavu

Research Center in Computer Science
University "Stefan cel Mare"of Suceava, Romania
pentiuc@eed.usv.ro, raduvro@yahoo.com

Abstract

The robot arm is an ongoing computer vision project for which many enhancements have been done during the last years, The aim of the paper is to present a system for controlling a robot arm, based on image processing and recognition. In the frame of this system, an important stage is the calibration. The presentation is focused on a new self-calibration algorithm based on video motion and color detection. The algorithm proves to be general enough to be employed in other similar computer vision scenarios where self-calibration might be useful. Calibration gets performed in a real-time having a low complexity per processed frame of *O(n*log(n))*

1. Introduction

In the Research Center in Computer Science from the University "Stefan cel Mare" of Suceava was developed several versions of a system for a robot arm control (called Hercules) using image processing and pattern recognition techniques. The control is represented by the robot's movements of grabbing and moving objects located on an operational table to user selected destination points.

Self-calibration is an important step allowing for controlling systems to auto determine their geometric parameters without external measurements, user conducted on-scene measurements or other ground truth calibration data. The advantages are obvious: more reliable scene measurements results obtained in less time and no user supervision required.

Robot self-calibration is not a novel idea and it proves more and more to be a requirement especially for mobile robots [1, 2, 4]. The situation is not a simple one either for static scenarios such as arm robots, robot platforms or other stationary robot devices, for which several approaches have been considered using forward motion analysis and stereo depth measurements [1], pose determination and use of different viewing angles [2], use of stereo perception [3].

A new approach for self-calibration of a video camera arm robot is proposed taking into consideration video motion and colour detection. Calibration gets performed in real time having a very low complexity of O(n*log(n)) where n is the video frame dimension. Moreover, the algorithm idea proves to be general enough to be extended for other computer vision scenarios where self-calibration may be employed.

Several enhancements with reference to the robot working mode in real-time are described such as region of interest processing and local histogram threshold selection.

2. The robot system description

Hercules, the robot, is a manipulation arm with a polar structure articled in 5 axes with pliers to catch objects with a maximum weight of 1 kg; the arm may perform actions in a hemisphere with a radius of 501 mm. A control box containing the electronic circuits that command the step by step motors of the robot arm; the box may be connected to a serial port of the computer. The control box permits the transmission of commands to the robot for positioning of its articulated pieces as well as it transmits to the computer information concerning the angles that these pieces make with respect to the axes.

The image processing module consists of a computer and a video camera centered on the working area. The control software system has two main components which are responsible for:

- acquisition and processing the images captured by the video camera and
- commands synthesis and transmission to the robot

In the elaboration process of the commands the calculus of the objects coordinates' is needed. The geometric parameters expressed in pixels will be converted in mm and angles in a polar coordinates system of the robot.

The functions of the image processing module are as follows:
- pre-processing of the video grabbed images
- contour detection and object identification and
- calculus of geometric parameters (all values are expressed in pixels).

These functions are accomplished in successive stages, the goals being the identification of the objects in the working area and their localization with respect to the coordinates system of the image (or the screen system).

The commands for the robot are issued from an image analysis process using coordinates' transformations from one system to another.

The control system makes use of three coordinates systems: the screen coordinates system; the working area coordinates system and the robot polar coordinates system (figure 1).

In order to determine the working area coordinates system, a rectangular shape is drawn on the table, one edge originating in and oriented along the X axis.

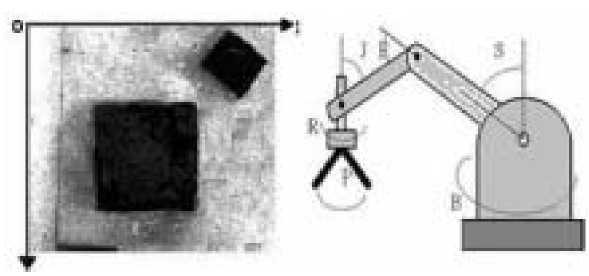

Figure 1: The image coordinates system Schematic and the polar coordinates system of the robot arm.

The module for the elaboration and transmission of the commands to the robot performs the following functions:

- conversion of the coordinates of the objects from the screen system to the metric units of the working area system
- conversion of the metric coordinates given in the working area system to the polar coordinates of the robot system.

A calibration process is necessary in order to accomplish the last conversion. Every time the relative position of the robot to the working area changed, the calibration program must be run again. The calibration program prompts the user to move the robot arm above the origin of the axis. For this position the program reads the values of the 5 angles. After that, the robot arm must be moved at any point above the X axis and the values of the 5 angles are read again. The parameters of the conversion functions are determined based on these 10 values and saved into a file on the disk. At the beginning of each execution the control system software reads these parameters from the file and uses them to compute new commands for the robot.

3. Image processing system with self-calibration

The accuracy of the commands generated by the control system software greatly depends on the accuracy of the maneuvers performed by the operator during the calibration process. Moreover, the calibration program is performed as an independent stage, initiated by the operator after had observed the robot's errors during the object grab-move task.

For these reasons the calibration program was replaced by a self-calibration stage, executed every time the control system software started. At the same time, a new version of the control system was designed, incorporating faster and more robust object-detection and also including a user friendly graphical interface.

The embedded version of the software system has two main components, namely:

- the configuration module responsible for the self-identification of the parameters used for coordinates conversion; this module determines the axes of the coordinates system and the working system scale constant based on the results of image processing and detection of a calibration red square shape label (width of 3 cm), attached to the robot base.

- the command module which includes real time video detection of the objects placed in the working area and robot arm control.

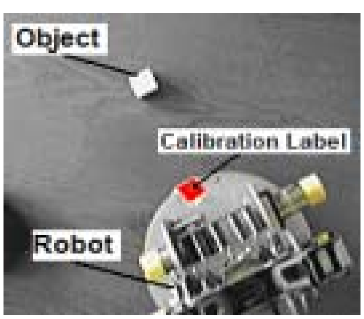

Figure 2: Top view of the robot arm. in vertical position, and the working area.

The object detection procedure supposes the existence of a minimum contrast between object colour and background colour. The background colour is non-homogenous (as it can be seen in figure 2). A technique based on background substraction as in [4] has been tried. The objects detection procedure is performed by applying a logical AND operation to the current video frame and an image mask determined during the calibration process.

Computing the threshold for the segmentation process using the histogram computed for a region of interest only proves to produce more robust object of interest segmentation as well as lower processing time. The video format used is RGB24, the dimension of the video frames 320x240 and the speed of processing of 25fps.

An important preliminary stage of each command generation is the self-calibration. In our case, calibration consists in determining the following parameters:

- the coordinates' axes of the 2D work field system

- the working scale depending on the height at which the video camera is mounted

The coordinates' axes of the working area are determined using two points: the weighted centre of the robot (x_g, y_g) and the centre of the calibration label (x_{red}, y_{red}) (figure 3). The equation of the first axis is determined using the coordinates of these two points whilst the equation of the second one is computed using the perpendicular condition on the first axis and by knowing that the origin is in the robot's weighted centre (1).

$$\begin{cases} d_1 : y = a_1 x + b_1 \\ d_2 : y = a_2 x + b_2 \end{cases} \quad \begin{cases} \begin{bmatrix} x & x_g & x_{red} \\ y & y_g & y_{red} \\ 1 & 1 & 1 \end{bmatrix} = 0 \\ a_1 \cdot a_2 = -1 \\ y_g = a_2 x_g + b_2 \end{cases} \tag{1}$$

The equations of the 2 axes are described using only 2 parameters as the special cases are trivial and are being treated separately (i.e. the 2 equations become $y = y_g$ and $x = x_g$).

The localization of the robot weighted centre is based on image processing and motion detection techniques. In order to label pixels in the image as belonging to the robot, the program commands a 45° rotation of the robot base (angle B in the figure 1) after which the base is immediately returned to the initial position. Motion detection takes place during this rotation. This procedure considers that the robot is the only moving object in the camera view and that it is entirely visible in the video frame (figure 2). Considering that the calibration gets performed one time only, prior to any actual commands, these conditions are pretty fair ones. The working flow of the calibration algorithm is given in figure 3.

Each video frame is first pre-processed by applying a median filter for noise reduction and two Sobel operators (2) for horizontal and vertical edge detection.

$$k_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad k_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{2}$$

The result is an image containing only the edges detected by the Sobel operators. Motion detection is performed by applying the difference between two pre-processed

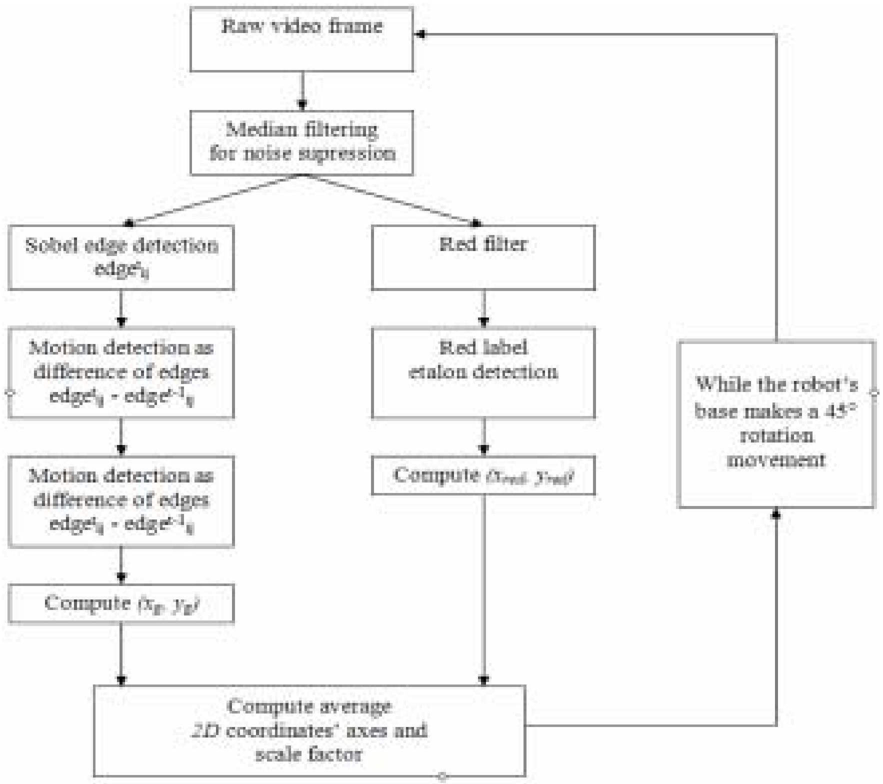

Figure 3: Workflow of the self-calibration algorithm.

consecutive frames considering that motion occurred at a pixel level if the difference is greater than a predefined threshold as in (3).

The result is a binary image with 1/0 levels coding for motion/non-motion. Subtracting edge images was preferred to a simple frame differencing as the difference of edge pixels provided robust results and gave a plus on accuracy as to the information on the actual motion.

$$d_{ij} = \begin{cases} 1 & if \left| edge_{ij}^{t+1} - edge_{ij}^{t} \right| > T \\ 0 & otherwise \end{cases} \quad (3)$$

where

- $edge^{t}_{ij}$ and $edge^{t+1}_{ij}$ are the edge images of 2 frames at consecutive moments t, $t+1$
- T is a predefined threshold for edge segmentation (manually determined after experimentation) and d_{ij} is the final binary differences image

The centre of the robot is computed using the binary edge differences image according to (4).

$$\begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} \dfrac{1}{\sum\limits_{d_{ij}=1} 1} \sum\limits_{d_{ij}=1} j \\ \dfrac{1}{\sum\limits_{d_{ij}=1} 1} \sum\limits_{d_{ij}=1} i \end{bmatrix} \quad (4)$$

For the detection of the calibration label, a red square, a simple red colour filter is applied to the original non edge processed video frame (5) that leads to a 2nd binary segmented image.

$$Output(r,g,b) = \left[r > 1.2 \cdot g \ AND \ r > Threshold \right] \quad (5)$$

The resulting image contains only the objects having a certain amount of red, objects being represented as connected components of pixels. In order to accurately detect the calibration label, the results are filtered according to the following criteria: surface restriction (objects' area values must be in a specific interval) and distance restriction (the label is the closest object to the robot weighted centre). In addition to these, for all the objects that passed the above restrictions, a circularity measure is computed (6) and by knowing that the label has a square shape, the object having the circularity closer to $\pi/4$ is selected and designated as the etalon label for all further reference (7)

$$C_i = \frac{4\pi A_i}{P_i^2} \quad (6)$$

where A_i/P_i are the area/perimeter of i*th* object.

$$red\_label = Object_j \ if \left| C_j - \frac{\pi}{4} \right| = \min_i \left\{ \left| C_i - \frac{\pi}{4} \right| \right\} \quad (7)$$

The scaling factor (depending on the height at which the video camera is mounted) is computed by division of the geometrical parameters of the label expressed in pixels, and the same parameters expressed in mm (the measures in mm are considered in our system ground truth).

The total amount of processing for a single video frame during the calibration phase is presented in table 1. Summing up the complexities of the various stages we finally obtain the calibration process complexity of ***O****(n*log(n))* where n is the dimension of the video frame.

Table 1: Complexity order of the various calibration stages

Median filter	***O****(n)*
Sobel edge detection	***O****(n)*
Motion detection	***O****(n)*
Red filter	***O****(n)*
Red label detection	***O****(n*log(n))*
(x_g, y_g), (x_{red}, y_{red}) computation	***O****(n)*

Objects segmentation is done based on image processing techniques being performed only in a semicircular region called the region of interest where the robot performs all the action.

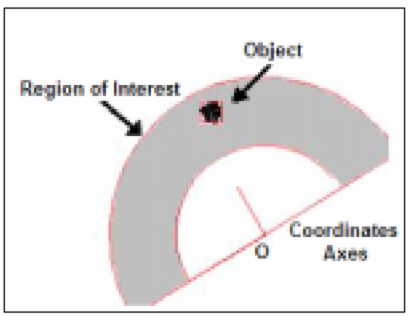

Figure 4: Region of interest computed as a Boolean mask for time computing reducing.

For this region of interest (figure 4), a Boolean mask is built at the end of the calibration stage based on the position of the robot weighted centre, the scaling factor, and the radius of action of the robot.

This mask is used as a first step AND operation performed on the current video frame which limits all the further processing (histogram computing, segmentation, object detection) for the region of interest. Because this region has an area representing about 1/5 of the entire image, an important reduction of processing time is obtained.

4. Conclusions

A simple self-calibration method was presented using video motion and colour detection technique. The total complexity of the algorithms per video frame is ***O****(n*log(n))* where n is the dimension of the frame.

The method can be employed for other computer vision tasks that require a self-calibration stage as it is based on simple principles: etalon colour detection and robot detection by means of motion.

Together with the proposed self-calibration method a few enhancements on the Hercules robot system are described that allow for real time image processing at 25fps. These enhancements are based on computing a circular region of interest implemented by the use of a Boolean mask. An unsupervised learning stage as in [6] and [7] may be useful for robust object recognition. Further developments may take into account the results presented in [8] and [9].

5. References

[1] Daniel Stronger, Peter Stone, “Simultaneous Calibration of Action and Sensor Models on a Mobile Robot”, *IEEE International Conference on Robotics and Automation, April 2005, Barcelona, Spain*

[2] Rodney A. Brooks, Anita M. Flynn, Thomas Marill, “Self Calibration of Motion and Stereo Vision for Mobile Robot Navigation”, *MIT Artificial Intelligence Laboratory, AI Memo 984, Cambridge, MA, USA*, 2000

[3] Yan Meng, Hanqi Zhuang, “Self-Calibration of Camera-Equipped Robot Manipulators”, *The International Journal of Robotics Research, SAGE Publications Ltd, London, UK,* Vol. 20, No. 11, 909-921, 2001

[4] N. Roy, S. Thrun, “Online self-calibration for mobile robots”, *Proceeding of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2292—2297, Detroit, 1999.

[5] Cerlinca, Tudor Ioan, “A Distributed System for Real Time Traffic Monitoring and Analysis”, *Advances in Electrical and Computer Engineering, Suceava, Romania* ISSN 1582-7445 - No 2, pp. 82-86, 2004

[6] Andrew Webb, “Statistical Pattern Recognition”, *2nd Edition, New York, John Wiley & Sons*, 2003

[7] Ștefan-Gheorghe Pentiuc, ‘Application of Pattern Recognition in Automatic Diagnosis” (in Romanian), *Ed. Tehnică, Bucureşti*, 1997

[8] Leonid Timchenko, Yuriy Kutaev, Volodymyr Kozhemyako, Alexander Gertsiy, Andriy Yarovyy, Nataly Kokryatskaya, “Method for Processing of Extended Laser Paths Images”, *Advances in Electrical and Computer Engineering, Suceava, Romania* ISSN 1582-7445 - No 2, pp. 66-78, 2003

[9] Zoran H. Peric, Bane V. Vasic, “Optimal Number of Phase Divisions in Polar Quantization”, *Advances in Electrical and Computer Engineering, Suceava, Romania* ISSN 1582-7445 - No 1, pp. 5-10, 2003