"VISION CONTROLLED PICK AND PLACE OF MOVING OBJECT BY 3R ROBOT"

Harika Nanduri[1], Dr. Manoj Soni[2],

[1] Master's Student, Indira Gandhi Delhi Technical University For Women, Delhi, India
[2] Associate professor, Indira Gandhi Delhi Technical University For Women, Delhi, India

ABSTRACT

Vision is a quintessential sense for any system, be it human system or a mechanical system. Industrial automation involves robotic arms for performing various factory operations, pick and place operation being their basic and most common application. Performing this pick and place operation in real time with vision sensors is a step deeper into automation. Motion of the object adds further complexity to the problem. This paper presents a 3R robot in eye-to-hand orientation for the application of pick and place of a moving object using vision system. The methodology consists of real time video extraction, object detection using the motion of object as classification criterion and point based tracking to predict future state of the moving object using Kalman filter in MATLAB. The tracked estimate is transferred to the robot using Arduino platform. A 3R robot with three degrees of freedom is subjected to the implementation of the methodology. A USB webcam is used to provide the visual information to the computer, where image processing is done on the captured video and object is detected and the future position of the object is predicted using Kalman filter algorithm. The robot is controlled using Arduino IDE to reach the predicted position and grasp the object with maximum accuracy. The paper emphasizes on the control of 3R robot with visual parameters. That is established by integrating the computer vision techniques with mechanical structure. The Arduino platform is integrated with MATLAB to implement the technique on the 3R robot and tested to reach the final position with satisfactory accuracy.

Keyword: *3R robot, Arduino, eye-to-hand, Kalman filter, Object tracking.*

1. INTRODUCTION

Computer vision has been a very active field of research since many years. The use of vision sensors in robotics has a long history. With the advent of technology, the manufacturing systems are employing robotic systems architecture for increase in productivity of the production line due to reduction in human interference hence increased accuracy and thereby reduced losses. The robotic manufacturing systems use robots to perform tasks and operation on the product. The positioning, orientation and scheduling of the workpiece at a particular workstation must be accurate in order to produce an error free product. If the position is even slightly disoriented, then it may cause the finished product to be faulty. Due to this large portion of the material may be wasted, which causes huge losses to the manufacturer. To reduce such losses often different tools such as jigs, fixtures, job holders etc are used to ensure that the job reaches the workstation at desired position. This is a traditional method of handling orientation problems in the robotic manufacturing setup. For the pick and place operation in industries, it becomes quite expensive to setup an environment exclusive to a particular part type and the conveyer has to be stopped at regular intervals for performing operations on the part. This increases the cycle time, which in turn reduces the productivity which is economically hazardous. The use of vision sensors in robotics has a long history.

1.1 Literature Review

The earlier methods were based upon open loop techniques of ‘look’ and ‘move’ concepts. By the end of 70’s systems capable of 10Hz servoing and 3D position control for tracking, seam welding and grasping were already demonstrated. Earliest reference can be taken that from Shirai and Inoue [8] who in 1973 explained how visual feedback can be used for correcting robot position to increase accuracy of the task. In this they used visual feedback to place a square prism in a square box using a 7dof manipulator.

Object detection is the process of finding instances of real world objects from the images or videos. Object detection algorithms typically use feature extraction for recognizing objects. A reliable object detection system is important for a wide variety of applications such as image classification, tracking, surveillance, and other specific application devices. Object detection can be done by using various object detection algorithms and softwares. Even for object detection a lot of work has been done in this field. A large amount of work has been done in this field using the sliding window approach, in which the object detection is achieved by increasing the search window size gradually. . Inspired by SIFT (Scale Invariant Feature Transform) [7], Dalal and Triggs [14] proposed to use HOG for the feature extraction step and they used SVM (Support Vector Machines) for the classification step. Many further studies improved the techniques for object detection.

When the object is considered in motion tracking is to be done to extract data from the video. The main objective of object tracking is to detect the desired object in the video and separating it from the rest of the environment and keeping a track on its position and orientation in the consecutive video shots i.e. frames. This has been an active research topic in computer vision field. A review on different tracking mechanisms has been presented in [1]. According to that, numerous tracking methods have been proposed in the past which primarily differ on their approaches for selection of the type of object being modeled, the image features opted for object modeling, environment in which the object is present and how the motion, appearance and shape of the object are being modeled. There has been great work done for tracking humans using articulated object tracking models presented in Aggarwal and Cai [2], Gavrilla [3], and Moeslund and Granum [4]. Lipton in [5] presented a simple method based on combination of temporal differencing and image template matching which provided highly satisfactory results even in the presence of clutter and enabled good classification avoiding special algorithms like Kalman filter for tracking. Isard and Blake [6] proposed a stochastic algorithm for highly robust tracking of agile motion in clutter environment superior to that attainable by Kalman filter. Although the methodology presented in this paper uses kalman filtering technique for object tracking as the setup environment is assumed to be clutter free.

Stauffer and Grimson presented in [9] a method of segmentation by modeling each pixel as a mixture of Gaussians and using an on-line approximation to update the model. This method proved to provide a solution for stable outdoor tracking with variations in lighting.

Liu in [10] proposed method for segmentation by background subtraction between reference frame and current frame. Similarly many other researchers developed the modeling techniques for object tracking. As mentioned earlier the tracking technique used in this paper is Kalman filtering technique. This technique was first introduced by Kalman in his paper [11] in 1960 for obtaining a recursive solution for discreet data linear filtering problem. With the advancement in digital computing techniques, kalman filtering has been a subject of extensive research since then majorly in the area of autonomous and assisted navigation. Introduction to the general idea of Kalman filter was presented by Maybeck [12] while Sorenson [13] presented a more detailed discussion on the topic with historical developments in the field. More extensive research work in the field was presented in [14], [15], [16] and [17].

1.2 Organization of the paper

The first section of paper presents an outlook on the motive of the paper and the research that has been done in the field. The second section is composed of the methodology behind this paper and the details of each step in obtaining the desired result. The third section describes the details of the experimental setup used. Fourth section presents the results obtained and the final section contains the conclusion and future scope of this paper.

2. METHODOLOGY

The methods followed in this paper are selected to integrate the computer vision system with the mechanical system for achieving the specified task of pick and place. MATLAB software has been used for the implementation of algorithm. MATLAB is efficient software for various numerical and scientific computations. MATLAB consists of various toolboxes for assistance in code generation. Image processing techniques are the heart of any computer vision project. MATLAB is used in this paper for implementing the image processing algorithms on the video sequences. The flow chart of the operations performed for object tracking in MATLAB can be represented. The flow chart shows the steps that have to be followed for implementation of the concept presented in paper. The video captured is processed and analyzed in order to track the moving object in real time.

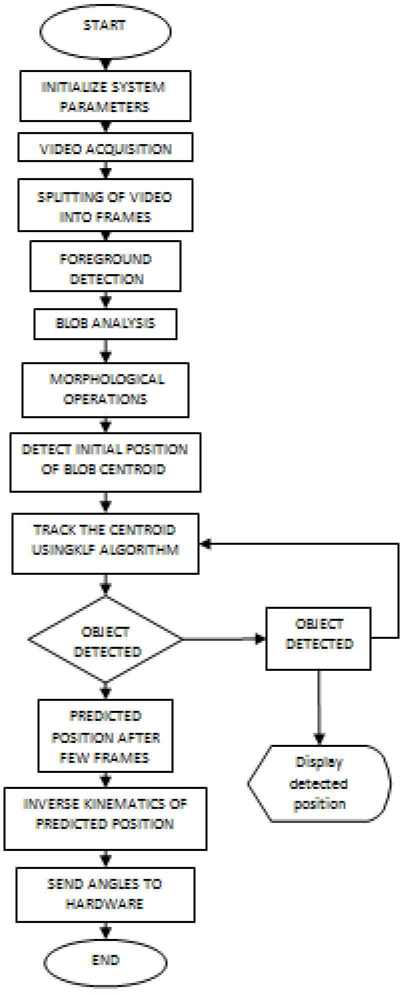
START
INITIALIZE SYSTEM PARAMETERS
VIDEO ACQUISITION
SPLITTING OF VIDEO INTO FRAMES
FOREGROUND DETECTION
BLOB ANALYSIS
MORPHOLOGICAL OPERATIONS
DETECT INITIAL POSITION OF BLOB CENTROID
TRACK THE CENTROID USINGKLF ALGORITHM
OBJECT DETECTED
OBJECT DETECTED
PREDICTED POSITION AFTER FEW FRAMES
Display detected position
INVERSE KINEMATICS OF PREDICTED POSITION
SEND ANGLES TO HARDWARE
END

Fig-1: Flow chart of the methodology

2.1 Image Processing

i. Image segmentation

The first step is to detect the object of interest. The object detection is done by background subtraction or foreground detection. MATLAB consists of 'ForegroundDetector' function in its computer vision toolbox, which performs segmentation on the successive frames to obtain a background and a foreground. Background is defined by the pixels which remain constant in every frame while foreground comprises of the image pixels which keep on changing its position. This is called motion based object detection as explained in second chapter. The computer may detect a number of moving pixels in the frames due to external noise; to select a particular object of interest blob analysis is performed.

To prevent the environmental noise from interfering with the segmentation process, thresholding is done. Thresholding involves providing a cutoff value above which the segmentation is to be performed. Threshold value is chosen such that the object blobs are detected efficiently. In this project threshold value of 5000 is used for segmentation.

ii. Blob analysis:

The foreground is detected as cluster of pixels which are changing their location. These clusters are called as blobs. Blob analysis involves computing the properties of blobs from the frames. A number of properties can be computed from the blobs like its major axis, area, dimensions, centroid and shape etc. these properties can be calculated using the 'regionprops' function in MATLAB. Also 'BlobAnalyzer' tool is available in computer vision toolbox. Blob area property is used to select the desired blob. The object is assumed such that having a minimum blob area of 300 pixels.

2.2 Object tracking using Kalman Filter

Point based tracking involves analyzing the trajectory of a single point. The centroid of the blob as calculated in previous step is chosen as the point of interest. Centroid of the blob is the centre of mass of blob i.e. object in our case. Kalman filter algorithm as explained in previous chapter is a linear data filtering method. It is selected for this paper as the object is assumed to be moving at a constant velocity thus its motion is linear. Kalman filter is a very efficient algorithm for obtaining the best possible estimate even with limited number of data inputs available. It is a low cost and low time consuming algorithm. MATLAB consist of Kalman filter function for directly implementing the algorithm without the need of writing any mathematical equations in the program.

Kalman filter is a powerful tool for combining information in the presence of uncertainty. It is an ideal method for linear continuously changing systems. A lot of literature is present on Kalman filter which makes it quite difficult to understand.

This technique was first introduced by Kalman in his paper [11] in 1960 for obtaining a recursive solution for discreet data linear filtering problem. With the advancement in digital computing techniques, Kalman filtering has been a subject of extensive research since then.

Kalman Filter Equations: The state of the object $\overrightarrow{x_k}$ is represented as set of its position vector and velocity vector.

$$\overrightarrow{x_k} = \begin{bmatrix} \vec{p} \\ \vec{v} \end{bmatrix} \qquad (1)$$

$$\hat{x}_k = F_k \hat{x}_{k-1} + B_k \vec{\mu}_k \qquad (2)$$

$$P_k = F_k P_{k-1} {F_k}^T + Q_k \qquad (3)$$

Equation (2) describe that the new best estimate $\hat{x}_k$ is the prediction made from previous best estimate $\hat{x}_{k-1}$ with correction B_k for known external influences.

$$\hat{x}'_k = \hat{x}_k + K'\left(\vec{Z}_k - H_k \hat{x}_k\right) \qquad (4)$$

$$P'_k = P_k - K' H_k P_k \qquad (5)$$

Where $K' = P_k H_k^T (H_k P_k H_k^T + R_k)^{-1}$

$\hat{x}'_k$ and P'_k represent the final state estimate of the object considering the measurement values from the sensor and its covariance matrix respectively. These equations predict the state with updating the uncertainty in measurement and prediction.

2.3 Inverse Kinematics for calculating Robot pose

These pixel coordinates are converted into real world coordinates of the object by applying geometric transformation methods. The coordinates of the object are converted from camera frame to robot base frame using frame transformation matrix generated based on the arrangement of the system and the coordinates are obtained in terms of the robot base frame. The angles required to align the robot in such a way that end effector reaches that point are calculated using the inverse kinematics. Inverse kinematics is a more challenging problem as compared to forward kinematics of robot. The angles are calculated using the geometric computation method for solving inverse kinematics solution.

2.4 Implementation on robot

Arduino-MATLAB interfacing is necessary for integration of computer vision with servo control. All the communication with the 3 robot is done serially through Arduino by storing values in this Arduino object. Thus angles calculated from the final position estimate are sent to the robot and the three motors align themselves to reach the object.

3. EXPERIMENTAL SETUP

The robot manipulator used for implementation of the methodology is a 3-dof articulated robotic arm. An articulated robot is a type of robot configuration in which all the robot joints are revolute similar to a human arm. In the setup, arm motion is controlled by three servo motors attached at the joint using Arduino IDE.
The robot manipulator used for implementation of the technique is a 3-dof articulated robotic arm. The arm motion is controlled by three servos attached at the joint using Arduino. The robotic arm constructed is a low cost mechanism made up of 1mm aluminum strips. The strips are used to make light weight clamps which are holding the servo motors and enabling the rotation at each joint.

3.1 Motors

Servo motors were used for the actuation purpose. In servo motor, a suitable motor is coupled to a sensor which provides it feedback signal for the control of its motion and final position. Hence servo motors have high accuracy i.e. servo motor is any motor working in a closed loop. The input signal to a servo motor is a PWM (pulse width modulation) signal which makes it align at angles ranging from 0^0 to 180^0. The servo motors used in the robot are MG995 high speed metal gear dual ball bearing servos manufactures by Tower-Pro of 10 kg-cm torque rating.

3.2 Vision Sensor

For providing the visual feedback to the robot, a visual sensor is required. A vision sensor may be a laser scanner, SONAR or camera. A camera is the most suitable option for vision sensing in robotics. With camera basically a computer vision system can be setup in two configurations- eye-in-hand configuration and eye-to hand configuration. In eye in hand configuration, the camera is attached to the end effector of the robot. While in the later one camera is taken stationary. The robot was configured in eye-to hand setup. Camera was placed stationary parallel to the base of robot. The camera employed was Microsoft LifeCam Show™, 2.0 MP webcam. The camera resolution of 800x600 pixels is used.
As stationary camera is used, the environment is needed to be calibrated. The system is setup such that the camera is placed near to the robot. The field of view of camera is measured to be around 26 cm along horizontal axis and 20 cm along the vertical axis. The object is placed at a distance of 13 cm from the robot so that the object remains in the robot workspace.

3.3 Arduino IDE

Arduino UNO is the most suitable project friendly board in the arduino family. This board is utilized in this project. The board as shown in the figure consists of several ports. The main controller used in the board is ATmega controller. Arduino UNO consists of 6 analog pins (from A0 to A5) for reading analog inputs from the sensors and converting them into digital signals. It consists for 14 digital IO pins from 0 to 13. Some of the digital pins may also be used for PWM. Other pins present on the board are power supply pins (5 V or 3.3 V), ground pins, Aref pin. Reset button, USB port for connecting the Arduino to IDE and a power supply jack for providing external power supply are also present on the board.

3.4 Arduino-MATLAB Interface

The Arduino – MATLAB interfacing is done with the help of Arduino support package available at the Mathworks official website. After connecting Arduino to the computer, the interface program given in the support package is loaded in the Arduino and Arduino is created as a variable object in MATLAB. Connection is established with the Arduino board and values can be communicated to Arduino using MATLAB.
Arduino MATLAB interfacing is crucial for implementation of the methodology on the robot. It is the actual basis for integration of computer vision with mechanical structure. It helps in controlling the robot with the help of visual feedback which is the sole purpose of this paper.

4. RESULTS

The proposed methodology is implemented on a 3R robot with 3 degrees of freedom. The following limitations have been set for implementation of the technology on hardware:

i. The environment is stationary with minimum clutter.

ii. The object is moving with a constant velocity.

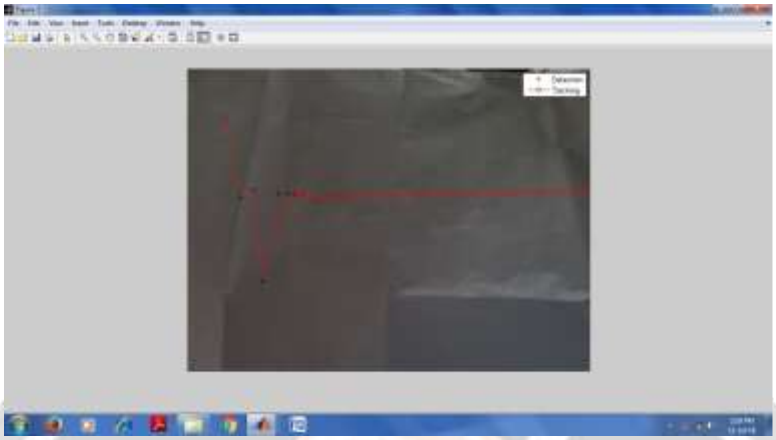

Fig-3 Tracked Trajectory of the object

Figure 2 presents a screenshot of the figure obtained on implementing the algorithm. The figure shows the trajectory of the path tracked using Kalman filter algorithm. The '+' represent the locations which were actually measured i.e. detected from the scene and red circles represent the predicted locations of the object.

Fig-4 Object in final position

Figure-4 represents the final orientation of the robot to pick up the object from predicted location.

5. CONCLUSION

This paper presents a technique to control a 3R robot with 3 degrees of freedom using the visual information captured from a USB camera to detect track and pick up an object moving at constant velocity. The robot is configured in eye-to-hand orientation. The advantage of using stationary camera as compared to a camera mounted on the end effector (eye-in-hand orientation) is that no compensation is needed for the errors due to the camera motion.

The motion of the object is the criteria used for the detection of object. Point based tracking method using Kalman filter is employed to predict the future position of the object using the current position. Kalman filter is a robust filtering technique which predicts and corrects the next state of the object using the present state measurement and previous estimates. Kalman filter works converges to the next position estimate with maximum accuracy even in the presence of high uncertainty. MATLAB software is used as the medium to generate the algorithm for this methodology. The methodology is tested on a self made 3R robotic arm using Arduino-MATLAB interfacing. The experimental results show satisfactory accuracy in computing the estimated point and reaching it in sufficient time to grasp the moving object.

The experiments show that the methodology is successful in tracking the object and picking it up with satisfactory accuracy. Small error occurs due to the system backlash and sometimes due to the inaccurate detections.

6. REFERENCES

[1] Alper Yilmaz, Omar Javed, and Mubarak Shah. *'Object tracking: A survey'*. Acm Computing Surveys (CSUR), 38(4):13, 2006.

[2] Aggarwal, J. K. And Cai, Q. 1999. *'Human motion analysis: A review'* , Comput. Vision Image Understand. 73, 3, 428–440

[3] Gavrila, D. M. 1999. *'The visual analysis of human movement: A survey'*, Comput. Vision Image Understand. 73, 1, 82–98.

[4] Moeslund, T. And Granum, E. 2001.*'A survey of computer vision-based human motion capture'*, Comput. Vision Image Understand. 81, 3, 231–268.

[5] Alan J Lipton, Hironobu Fujiyoshi, and Raju S Patil. '*Moving target classification and tracking from real-time video'*, Applications of Computer Vision, 1998. WACV'98. Proceedings, Fourth IEEE Workshop on, pages 8–14. IEEE, 1998.

[6] M. Isard and A. Blake. '*Contour tracking by stochastic propagation of conditional density'*, Proceedings of European Conference on Computer Vision 96, pages 343–356, 1996

[7] Lowe, D., '*Distinctive image features from scale invariant keypoints'*, International Journal of Computer Vision (IJCV), 60, 91-110, (2004)

[8] Y. Shirai and H. Inoue. '*Guiding a robot by visual feedback in assembly tasks'*. Pattern Recognition, 5, pp.99-108, 1973

[9] Chris Stauffer and W Eric L Grimson. *'Adaptive background mixture models for real-time tracking'*, Computer Vision and Pattern Recognition, 1999. IEEE Computer Society *Conference on.*, volume 2. IEEE, 1999.

[10] Ya Liu, Haizhou Ai, and Guang-you Xu, *'Moving object detection and tracking based on background subtraction*', In Multispectral Image Processing and Pattern Recognition, pages 62 66. International Society for Optics and Photonics, 2001.

[11] Kalman, R. E. 1960. "*A New Approach to Linear Filtering and Prediction Problems*," Transaction of the ASME—Journal of Basic Engineering, pp. 35-45 (March 1960).

[12] Maybeck, Peter S. 1979. *Stochastic Models, Estimation, and Control, Volume 1*, Academic Press, Inc.

[13] Sorenson, H. W. 1970. *"Least-Squares estimation: from Gauss to Kalman*," *IEEE Spectrum*, vol. 7, pp. 63-68, July 1970.

[14] Brown, R. G. and P. Y. C. Hwang. 1992. '*Introduction to Random Signals and Applied Kalman Filtering, Second Edition'*, John Wiley & Sons, Inc.

[15] Gelb, A. 1974. '*Applied Optimal Estimation*', MIT Press, Cambridge, MA.

[16] Grewal, Mohinder S., and Angus P. Andrews (1993). '*Kalman Filtering Theory and Practice*'. Upper Saddle River, NJ USA, Prentice Hall.

[17] Lewis, Richard. 1986. *Optimal Estimation with an Introduction to Stochastic Control Theory*, John Wiley & Sons, Inc.