

VISUAL CONTROL OF A ROBOTIC ARM

GEORGE CARTER, M. OSMAN TOKHI AND OUSMANE A. OUMAR

School of Engineering, London South Bank University, London, UK.

Abstract. Integration of robot control with computer vision is an emerging technology. The ability to capture a real world scenario, filter out a specific object, and then localise it with respect to a specific point opens up many avenues for future technologies. In this paper, computer vision is integrated with a 6-DOF robotic arm. The aim is to develop a methodology for controlling a robotic arm through visual interaction. The investigations carried out include mathematical and geometrical representation of the physical manipulator, object detection utilising a colour segmentation approach, establishing communications from the command line of the visual recognition program with an Arduino and development of a user orientated software package that integrates the two key objectives.

Keywords: Robotic arm, computer vision, object localisation.

1. Introduction

The aim for most robotic designs is to alleviate the need for human intervention; that is, for robots to perform tasks with more efficiency, accuracy and precision over a more sustained period. For a robot to execute a task in this manner, it needs to be equipped with human-like senses; somatosensation, audition, vision, balance and local body orientation. With the advancement of digital cameras, machine vision has become an interesting field of discovery. One of the key tools developed is OpenCV, which is an open-source computer vision library [1]. Scenarios utilising computer vision use a digital camera attached to the end-effector [2], a fixed [3] or a mobile robot [4]. The fixed approach has proved more accurate with its detection compared to mounting the camera on the end-effector and to the mobile robot due to error caused by vibration through their movement. To overcome this, stereovision has become a viable option, with algorithms developed to achieve synchronised vision using two cameras [5].

Two main image-processing routes are widely used due to their accuracy rates and amount of relevant literature available. The first technique utilises colour detection within the Hue, Saturation and Value (HSV) scale [6]. The HSV

parameters are varied to single out a specific colour before morphological operations are performed on the image to increase the accuracy of detection. The alternative to this is to use grayscale, which is a binary image equating the light intensity of each pixel with a particular shade of gray [7]. Similar to HSV, the BGR/RGB image is converted to grayscale in order to be processed. Once in this binary form an algorithm called canny edge detection is used to single out particular objects [8]. The detected object will have a white line around its edges, signifying the location. The major issue coming out of these two approaches are the accuracy of detection due to lighting and camera quality.

The work presented in this paper focuses on the control a 6 degree of freedom (DOF) robotic arm. In order to orientate the end-effector of a manipulator to a particular Cartesian coordinate, the inverse-kinematics approach is adopted, where the Cartesian space is transformed into joint variable space. There are several approaches available for solving the inverse-kinematics problem for a six-joint manipulator. These include inverse transform, screw algebra, dual matrices, dual quaternion, iterative and geometric approaches [9]. The possibility of extensive modelling of the approach chosen is imperative. The geometric approach was selected in this study as it enables modelling of all six joints individually and concurrently, providing extensive understanding of response of each joint. The work presented forms the basis of integrating machine vision with robot control. The concept could be extended for implementation in areas such as hazardous environments, exploration or explosive ordnance disposal.

2. Control Algorithm

Figure 1 shows the physical and mathematical representations of joint 1 of the manipulator. An external camera with a limited field of vision; 60° , is used, and to allow maximum coverage of the manipulator's workspace, the camera is mounted, at a 45° angle and located approximately 38cm to the left of the base of the robot.

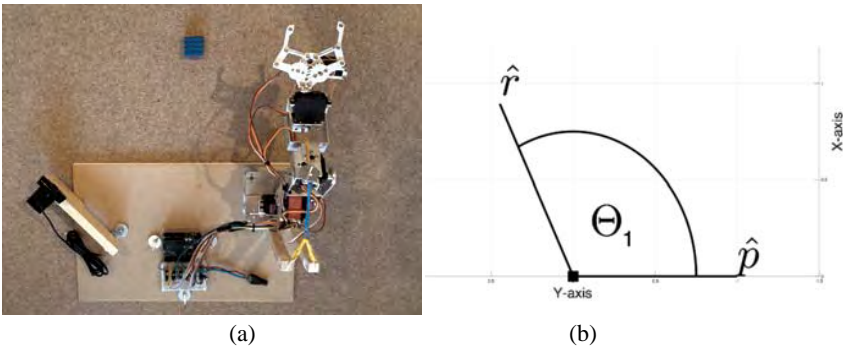


Figure 1: (a) Physical representation of joint 1. (b) Mathematical representation of joint 1.

The geometric approach to solving the inverse kinematics problem for a six-joint manipulator produces several solutions depending on the desired orientation of the arm. In order to minimise the number of solutions available, enabling full robot autonomy, the six joints are broken down into sections. Each section, in turn, is then analysed and an algorithm for its control is generated, along with a visual representation of the joint.

The purpose of joint 1 is to orientate the robot arm in the direction of the object. For the servomotor to move to a desired location, a joint angle must be specified. If the servomotor is already stationed at 90° and a new angle is specified, the servomotor would move from 90° to the new angle. This functionality means that the displacement angle of the object needs to be referenced to a fixed point, \hat{p} .

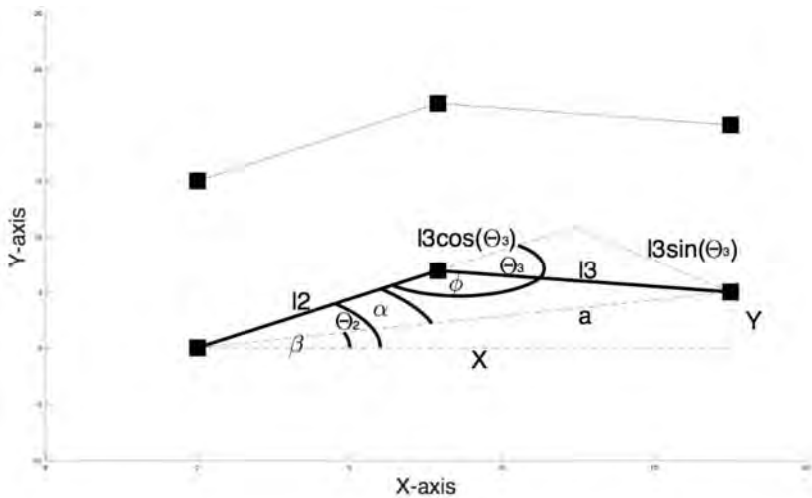


Figure 2: Mathematical & geometrical representation of joints 2 & 3

The final three joints would orientate the end-effector so that it is perpendicular to the face of the object it is grabbing. With this operation joints 2 and 3 would position the arm so that joint 4 is able to rotate the final joints to this objective. Figure 2 shows a geometric representation of joint orientations, where x and y are the desired coordinates of joint 4, l_2 and l_3 are the lengths of link 2 and link 3 respectively, a is the magnitude of x and y , θ_2 is the displacement angle of joint 2 from the x -axis, θ_3 is the displacement angle from the continuation of link 2 and link 3, ϕ is $\pi - \theta_3$, α is the difference of $\theta_2 - \beta$, where β is the displacement angle of a from the x -axis. The calculation for each parameter is as follows:

$$\phi = \text{atan2}(\cos(\phi), \sin(\phi)) \quad (1)$$

where,

$$\cos(\phi) = \frac{l2^2 + l3^2 - a^2}{2 \cdot l2 \cdot l3} \quad (2)$$

$$\sin(\phi) = \pm \sqrt{1 - \cos^2(\phi)} \quad (3)$$

$$\theta_3 = 180 - \phi \quad (4)$$

$$\beta = \text{atan2}(x, y) \quad (5)$$

$$\alpha = \text{atan2}(l2 + (l3 \cdot \cos(\theta_3)), l3 \cdot \sin(\theta_3)) \quad (6)$$

$$\theta_2 = \alpha + \beta \quad (7)$$

Therefore, the resultant servo joint angles are given as

$$\text{Joint2} = 90^\circ + \theta_2 \quad (8)$$

$$\text{Joint3} = 90^\circ + \theta_3 \quad (9)$$

3. Pinhole Camera Model

The pinhole camera model depicted in Figure 3 shows the approach for real world mapping onto an image plane for a calibrated camera at a single instance of an objects depth from the camera. F_c is the optical centre, from which all of the information from the real world is passed to the image plane. The distance from the optical centre to the principal point changes depending upon the real world value of the objects distance from camera, known as the focal length. Calibrating this enables rudimentary calculation of the objects distance from the focal point.

Figure 4 shows measured values of focal length as function of distance from optical centre. The blue line in Figure 5 illustrates how the focal length changed with respect to the distance of the object. As the object moved further away, the focal length increased, insinuating that it was increasing the distance from the optical centre to the principal point allowing more information from the real world to be captured. The line of best fit generated a function for the focal length as

$$f(z) = (0.2537 \cdot z) + 658.7 \quad (10)$$

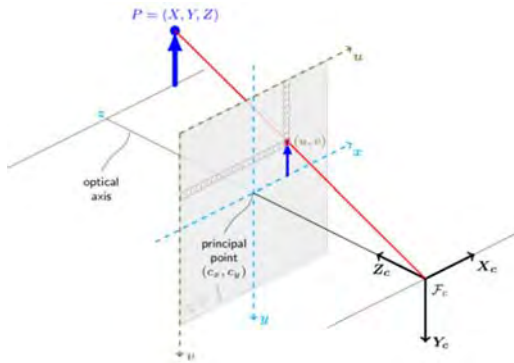


Figure 3: Pinhole camera model [10].

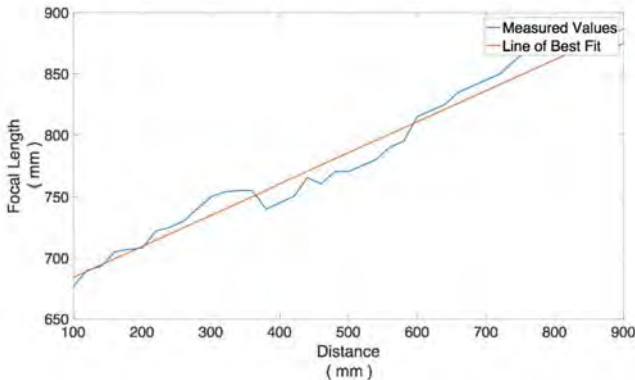


Figure 4: Focal length with respect to distance of object from optical centre.

4. Object Detection and Localisation

Figure 5 illustrates the procedure undertaken within the visual recognition algorithm. Utilising the colour segmentation approach, understanding the HSV scale was imperative. Hue defines the colour type; saturation defines the brightness and value the intensity of the colour. OpenCV enables user interactivity through graphical user interfaces. In this case, a trackbar was used to vary the values of HSV to visualise the filtering. Once filtering has taken place the threshold image undergoes morphological operations; erosion, dilation and Gaussian blur, removing any unwanted noise in the frame. After the object has been correctly filtered its contours and centre of mass are detected, with a bounding box and centre circle displayed to the user on the camera frame. The focal length is updated on each frame, increasing its precision, from which the x , y & z coordinates with respect to the camera frame are generated. These are then

rotated and translated with respect to the robot base in order for the joint angles to be calculated.

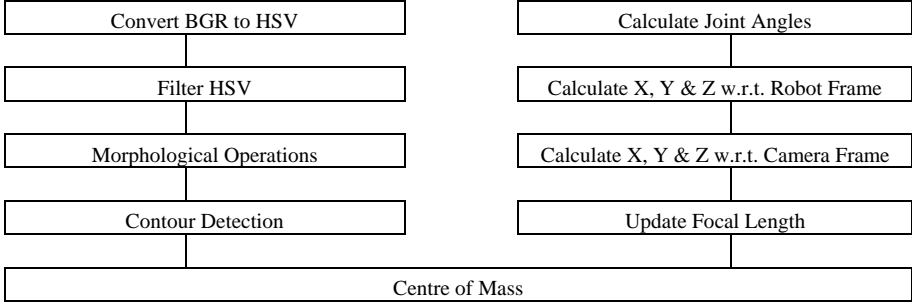


Figure 5: Visual recognition algorithm

Figure 6 demonstrates the visual recognition program detecting the yellow object; drawing a bounding box around it as well as its centre of mass, displaying its dimensions in pix-els and millimetres and its location in pixels as well as with respect to the robot base in millimetres. The left hand frame is the camera feed, the right hand frame is the threshold image, and the two lower trackbar boxes are used to vary the hue, saturation and value parameters and the object’s dimensions in millimetres.

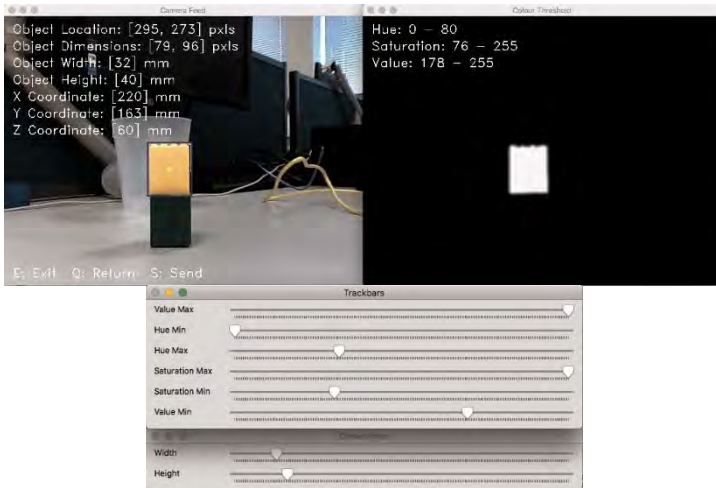


Figure 6: Visual recognition program detecting and localising object with respect to robot base.

5. Communication

The communications platform CoolTerm was chosen as the means of communication [11]. This piece of software accesses the USB port and sends a string of characters in ASCII format via TTL logic. At first, the TX-RX cables of the TTL were connected to the Arduino. With some manipulation of the SoftwareSerial library within the Arduino, the ASCII characters were converted back to decimal integer format, allowing a more complete design to digest and disseminate data effectively and efficiently ensuring the robot autonomously orientates itself to grab an object.

6. Results and Analysis

Object detection and localisation was achieved, yet with some error. The curvature of the lens assumed the object was closer to the centre than it actually was. However, as the curvature was uniform it could be modelled and an error reduction algorithm was generated to compensate for this undershoot. Implementing the error reduction algorithm the visual recognition program correctly detected and localised a block coloured object with respect to the robot base. The servomotor controlling the first joint produced some error due to the loading of the successive joints and object. The servomotor was undershooting from $45^\circ - 135^\circ$ and overshooting from $0^\circ - 45^\circ$ and $135^\circ - 180^\circ$. An algorithm was written to compensate for this under and overshooting. A communications platform was established utilising the software program CoolTerm. Data was sent and received as ASCII characters between the Arduino and the program correctly.

7. Conclusion

A visual approach for control of a 6DOF manipulator has been presented. The robot arm and camera have been mounted on one solid base with the camera positioned to maximise its field of vision. Although low quality, the camera has still correctly detected and localised the object on each trial. Reasonable accuracy in object detection and localisation has been achieved, and any errors caused by the quality of the camera have been mitigated by introduction of an error reduction algorithm. The visual recognition program has been shown to correctly detect and localise a block coloured object. The work presented provides a framework for further research into object detection and localisation, as well as precise robot control.

References

1. OpenCV Team. (2018). OpenCV Library [Online]. Available: <https://opencv.org/>

- 2.A. De Luca, M. Ferri, G Oriolo, "Visual servoing with exploitation of redundancy: An experimental study," International Conference on Robotics and Automation (ICRA 2008), 2008, pp. 3231-3237. doi: 10.1109/ROBOT.2008.4543703
3. Mussabayev RR. "Colour-based object detection, inverse kinematics algorithms and pinhole camera model for controlling robotic arm movement system," International Conference on Electronics Computer and Computation, 2015, pp. 1-9. doi: 10.1109/ICECCO.2015.7416879
4. J. Cernohorsky, P. Jandura, O. Mach. "Mobile robot localization and object description," 18th International Carpathian Control Conference, 2017, pp. 1-4. doi: 10.1109/CarpathianCC.2017.7970452
5. A. Tse, J. Saegnwipharat, K. Supayanant, "Controlling children-toy tank by using image processing technique," ICT International Student Project Conference (ICT-ISPC), 2016, pp. 73-76. doi: 10.1109/ICT-ISPC.2016.7519239
6. P. Khelbude, S. Shelke, "Real time iris controlled robot," Online International Conference on Green Engineering and Technologies (IC-GET), 2016, pp. 1-4. doi: 10.1109/GET.2016.7916648
7. M. Dragusu, A. N. Mihalache, R. Solea, "Practical applications for robotic arms using image processing," 16th International Conference on System Theory, Control and Computing (ICSTCC), 2012, pp. 1-6. [Online] Available: <http://ieeexplore.ieee.org/document/6379295/>
8. K. S. Fu, R. C. Gonzalez, C. S. G. Lee, Robotics: Control, Sensing, Vision, and Intelligence, International Edition, Book, Singapore: McGraw-Hill Book Co, 1987. ISBN 0-07-100421-1.
9. OpenCV Team. (2014). Camera Calibration and 3D Reconstruction [Online]. Available: https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html.
10. Roger Meier. (2018). Roger Meier's Freeware [Online]. Available: <http://freeware.the-meiers.org/>.