Automation of a wheelchair mounted Robotic Arm using Computer Vision Interface

Aditya Prakash Kaithvas[1], Pallavi N[2], Remya S[3]
Department of Computer Science Engineering
[1] Student, Atria Institute Of Technology
[2,3] Assistant Professor, Atria Institute Of Technology.

Abstract- *Assistive robotic devices have great potential to improve the quality of life for individuals suffering with movement disorders. One such device is a robot-arm which helps people with upper body mobility to perform daily tasks. Manual control of robot arms can be challenging for wheelchair users with upper extremity disorders. This research presents an autonomous wheelchair mounted robotic arm built using a computer vision interface. The design utilizes a robotic arm with six degrees of freedom, an electric wheelchair, computer system and two vision sensors. One vision sensor detects the coarse position of the colored objects placed randomly on a shelf located in front of the wheelchair by using a computer vision algorithm. The other vision sensor provides fine localization by ensuring the object is correctly positioned in front of the gripper. The arm is then controlled automatically to pick up the object and return it to the user. Tests have been conducted by placing objects at different locations and the performance of the robotic arm is tabulated. An average task completion time of 37.52 seconds is achieved.*

Keywords- Assistive Technology, Automation, Computer Vision, Image Processing, Robotic Arm, Vision Sensor

I. INTRODUCTION

Mechanical arms mounted on a wheel seat to a great extent advantage individuals experiencing chest area development issue, for example, Amyotrophic Lateral Sclerosis (ALS), Parkinson's ailment, Progressive Muscular Atrophy (PMA). They can upgrade the control capacities for electric wheelchair clients and make them feel increasingly autonomous. In any case, wheelchair mounted automated arms (WMRA) accessible in the market, for example, JACO Robot arm by Kinova Robotics [1] and iARM by Assistive Innovations [2] use remote controlled interface, for example, joystick to control the arm. This may challenge work for clients experiencing engine neuron ailments and much of the time it can expend a great deal of time and exertion to effectively get a handle on the item. They are additionally over the top expensive generally keep running into a huge number of dollars [1].

Numerous scientists have taken a shot at accomplishing an adaptable interface between the client and the mechanical arm like touch screen, joystick and realistic presentation [3][4] to choose the objective item. Studies have been performed in the region of human computer interface (HCI) to create assistive technology (AT) for controlling automated arms. Indika Pathirage at. Al. [5] executed brain computer interface (BCI) for getting a handle on an article by making a visual framework to choose an item from the scene picture. The examination demonstrated that utilizing BCI to choose an item can devour additional time and adds weakness to the client [5]. Hairong Jiang et. al. [6] performed object identification utilizing Speeded Up Robust Features (SURF) calculation and the outcomes were utilized for coarse situating of the arm and signal based acknowledgment was utilized for fine restriction. The greater part of the interfaces are self-loader and they require certain dimension of fixation and cooperation from the client. Their finish times are additionally generally more (around 200 to 300 seconds) [6].

Figure 1. Image of the robotic arm mounted on an electric wheelchair

Considering these disadvantages, a self-governing framework is planned that will help the electric wheelchair clients to get objects from a table or a rack easily in a short measure of time. The evaluated expense of the framework is around $1500, which is multiple times not exactly the monetarily accessible WMRA [1]. The objective of the examination is to build up a framework that can convey the ideal article to the client with insignificant to no supervision. This test is not quite the same as the current ones of every a way that the framework is intended to be client autonomous and can make sense of the situation of the article alone utilizing an item identification calculation and can total the assignment of grabbing the item and returning it to the client under one moment. The coarse area of the objective item is determined by utilizing picture got from the vision sensor situated over the arm and fine restriction is accomplished by utilizing a dream sensor close to the gripper. The acquired directions are nourished to the mechanical arm utilizing a sequential correspondence interface to move the arm the correct way.

II. SYSTEM DESCRIPTION

The proposed framework comprises of an automated arm with six degrees of opportunity, an electric wheelchair, a PC framework and two vision sensors. Figure 1 demonstrates the picture of the total framework. The automated arm is mounted on the left arm of the wheelchair. One of the vision sensor is set over the mechanical arm which catches the video of the item put on the rack. The situation of this vision sensor is static and the relative separation between the sensor and the base of the arm is known preceding the beginning of the test to convey precise outcomes. The rack is utilized to accomplish the objective of getting to objects at various statures. Another vision sensor is mounted over the gripper to calibrate the situation of the arm's gripper before it can pick the article. This vision sensor is dynamic and the position changes relying upon the situation of the objective article. A PC vision framework is built up that identifies different shaded articles set haphazardly superficially inside reach by the arm. Figure 2 demonstrates the execution of the proposed framework. The distinctive pieces of the framework are examined in detail underneath.

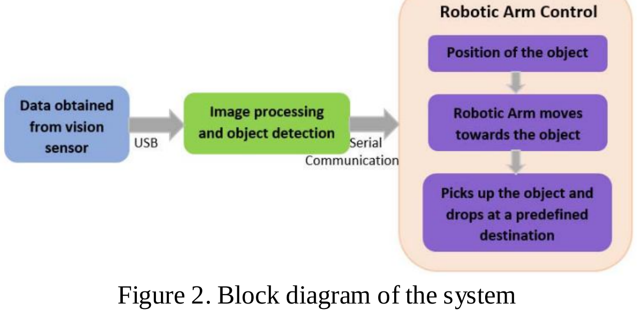

Figure 2. Block diagram of the system

ROBOTIC ARM

A Trossen Robotics PhantomX Reactor Robot Arm is assembled utilizing an Arduino perfect progressed microcontroller called Arbotix-M robocontroller [7]. The arm has eight AX-12A dynamixel actuators for controlling diverse pieces of the arm. Every servo has sensors to follow its speed, temperature, shaft position, voltage and burden. The servo engine at the base is utilized to move the arm the even way (left and right). The shoulder has double servo which moves the arm in the forward and reverse way. The elbow likewise has double servo which controls the here and there development of the arm. The wrist edge and wrist pivot have one servo each. The arm has a parallel gripper constrained by an AX-12 servo which can hold and discharge the item. The arm is controlled by a 12V 5amp power supply and is associated with a PC framework through a FTDI link. A sequential association is made between the PC framework and the mechanical arm so as to speak with the automated arm. This was accomplished by utilizing PySerial, a python sequential port access library. The sequential association has a baud rate of 38400. The information parcel which is 17-byte long was sent sequentially from the PC to the Arbotix-M robocontroller to control the development of the arm. The 17-byte information bundle incorporates the header (0xFF/255), X-hub organize, Y-hub facilitate, Z-hub arrange, wrist point, wrist turn, gripper, delta byte and check total. Every one of the servo engine can be constrained by differing the 17-byte information sent to the arm appropriately. A short postponement is presented after each sequential compose direction is performed to guarantee the gathering of the 17-byte information by the automated arm.

VISION SENSORS

The framework utilizes two vision sensors (USB webcams) to perform object location. A Logitech HD c920 webcam (vision sensor 1) is mounted over the mechanical arm confronting the rack situated before the arm. The vision sensor 1 catches the video of the arm and the rack continuously. Casings removed from this video are prepared and the position (X, Y) of the objective item is determined. This information is utilized for coarse situating of the automated arm. A robot VGA webcam (vision sensor 2) is mounted over the gripper utilizing a 200 mm gooseneck. The vision sensor 2 catches a nearby video of the objective article. Casings from this camera are caught simply after the arm is moved to the position shown by vision sensor 1. The vision sensor 2 is utilized to position the gripper precisely before the item with the goal that the article can be gotten accurately. The goals of the pictures in both the camera is 640 x 480 pixels and the webcam catches around 30 outlines for every second. Figure 4 demonstrates the preview of the picture caught by the two vision sensors.

COMPUTER VISION ALGORITHM

The mechanical arm is customized to move towards the situation of a particular hued object. The shading identification calculation is written in Python utilizing the OpenCV library. The vision sensor catches the continuous video of the mechanical arm and the item.

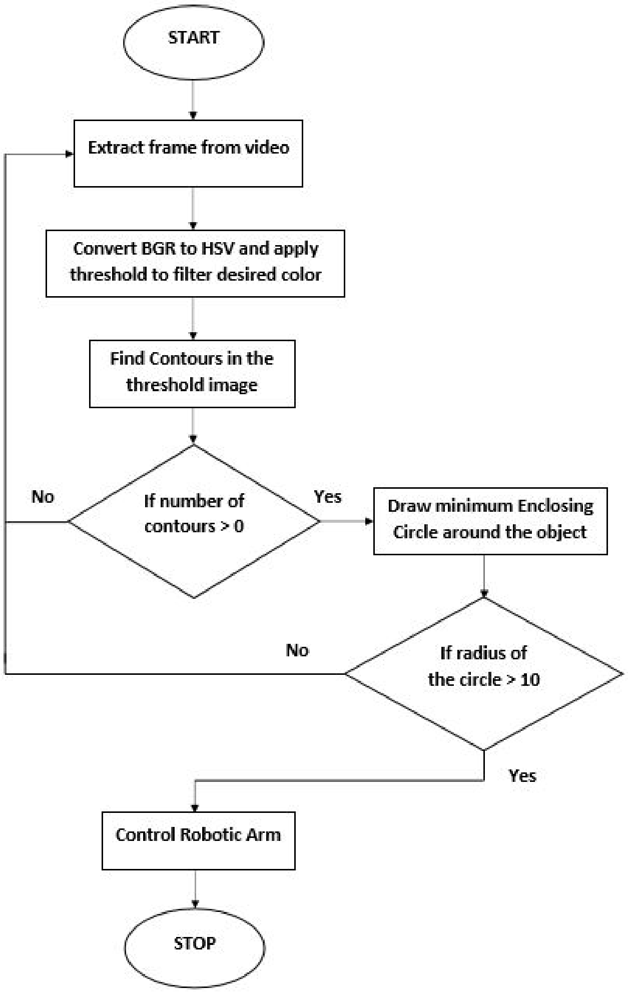

Figure 3. Flowchart for Computer Vision Algorithm

The video is separated edge by casing and every one of the casing is handled through a progression of ventures to recognize shaded articles in the casing. Each edge is changed over from BGR (Blue Green Red) picture to HSV (Hue, Saturation and Value) picture. A particular shading is separated by applying a lower and furthest limit edge to the HSV picture [8]. Diverse hues have distinctive scope of shade esteems. Subsequently by giving the fitting scope of shade esteems, distinctive hues can be distinguished at the same time. The investigation is performed with blue shading picked as the object of premium. The subsequent picture is disintegrated to evacuate the commotion present in the picture and enlarged to fill the holes in the picture. The edge picture is examined to check the nearness of shapes. In the event that a shape is found the situation of the form is found by ascertaining minutes [8]. A base encasing circle is drawn around the article and the focal point of the hover compares to the X and Y directions of the situation of the item. The shading location calculation is stretched out to catch information from two vision sensors and joined with the automated arm control code to play out the ideal assignment. Figure 3 demonstrates the flowchart execution of the computer vision algorithm.

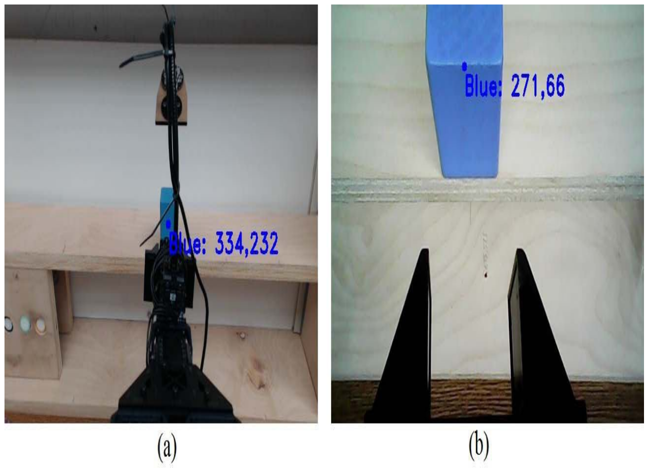

Figure 4. (a)Image captured by the vision sensor 1 (b)Image captured by the visioni sensor 2

III. EXPERIMENT AND RESULTS

The mechanical arm is worked in the Cartesian mode. At first the two vision sensors begin catching the video continuously. A casing is removed from the video caught by the vision sensor 1 and the X and Y position of the article is determined utilizing the shading recognition calculation. Before continuing with the arm control, a condition is verified whether the situation of the item is same as or near its past position (X ± 5). In the event that the condition is valid, it implies the article has not moved and henceforth the arm stays in the base position. In the event that it is false, it shows that the item has been moved to an alternate area and the arm is controlled to move towards the new area of the article. Along these lines it dispenses with the likelihood of the arm to move to a similar position more than once regardless of whether numerous casings are caught by the vision sensors. On the off chance that an article is recognized by the vision sensor, the arm is moved to the base position (a rearranged L) and the gripper is opened. The arm is then climbed or down contingent upon the estimation of the Y organize with the limit estimated at 300.

Arm Link programming given by Trossen Robotics is a basic realistic UI (GUI) to control the InterbotiX line of robot arms [9]. X organize scope of the arm differs from 212 to 812 and can be controlled physically utilizing the Arm Link programming. The pixel remove in the level pivot fluctuates somewhere in the range of 0 and 640 and is estimated utilizing the shading discovery calculation. The item is set at various positions before the arm and numerous readings of the X remove in pixels and the comparing X arrange of the arm are taken and plotted in the Desmos charting number cruncher. A best fit condition of a line is planned. The components 0.95 and 150 are determined utilizing experimentation technique to such an extent that the line created goes through a most extreme number of focuses plotted.

X position of the arm = (X object * 0.95) + 150 (1)

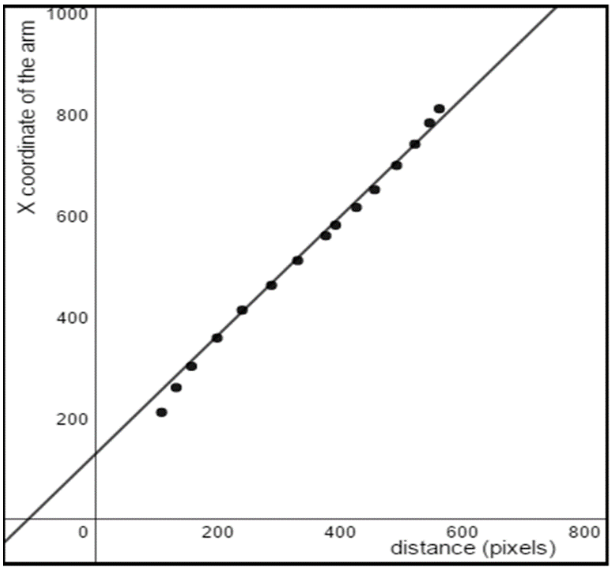

Figure 5. .X distance in pixel vs X coordinate of the robotic arm

Figure 5 demonstrates the diagram between X separate in pixel and X arrange of the automated arm. The arm is then controlled to move in the left or right course contingent upon the esteem acquired from the condition (1). Presently the mechanical arm is moved to a position which is exceptionally near the area of the objective item. There is a plausibility that the gotten area probably won't be the careful area of the item. Subsequently the vision sensor 2 is utilized to acquire the precise area of the item. The X esteem in pixel is verified whether it lies in the focal point of the casing. In the occasion it doesn't fall inside the predetermined range, the arm is moved in little strides in the left or right bearing until it achieves the middle position. In the event that the X esteem is as of now in the predefined extend, no direction is given to the arm. The article is grabbed by shutting the gripper and the arm at that point moves before the client to convey the item. Figure 6 demonstrates the flowchart usage of mechanical arm control. Table I and II demonstrates the X and Y position of the objective item acquired from the vision sensor 1 and 2. The time taken for the arm to get the article and convey it to the client is noted. It additionally indicates if the endeavor is a triumph or a come up short. Out of the 24 preliminaries performed with the items in the upper and lower level, 20 were fruitful in grabbing the articles accurately. This gives an exactness of about 83.33%.

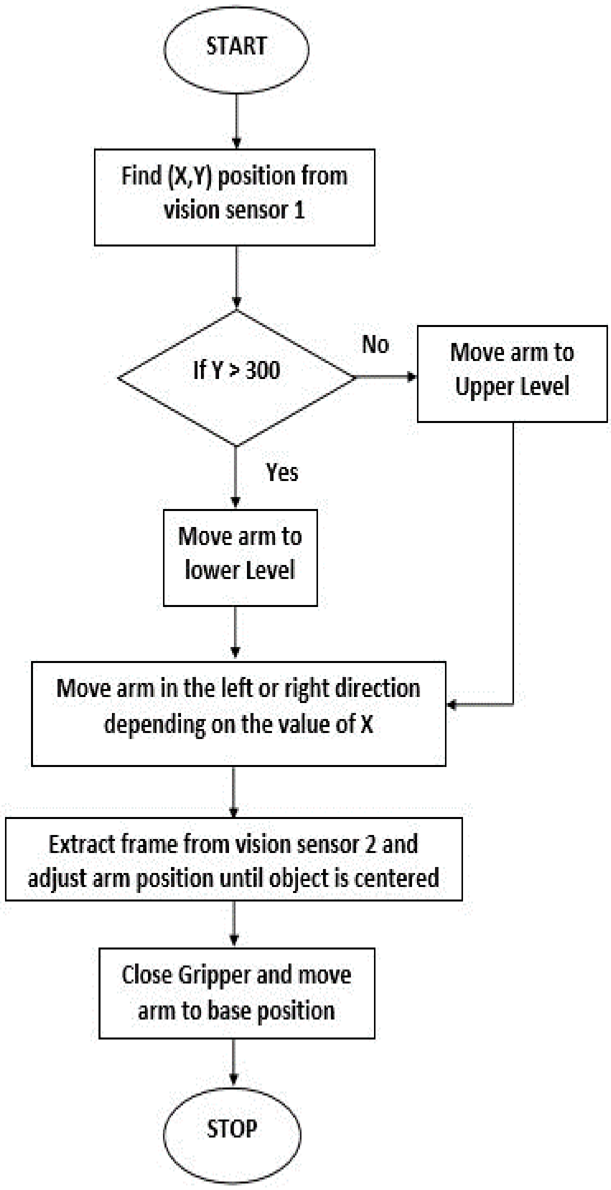

Figure 6. Flowchart for Robotic Arm Control

TABLE I
Upper level

Vision Senor 1 X, Y (in pixels)	Vision Senor 2 X, Y (in pixels)	Task completion time (in seconds)	Attempt Status (Success/ Fail)
259,219	69,179	39.51	Success
296,232	121,158	37.64	Success
326,222	324,82	37.8	Success
340,224	280,105	36.58	Success
356,256	143,139	35.79	Success
391,234	496,136	38.07	Success
411,230	590,166	37.65	Success
435,238	579,163	37.24	Success
615,244	590,166	37.65	Fail
630,229	547,114	35.28	Success
302,249	261,111	37.53	Success
248,242	63,145	37.66	Success

TABLE II
Lower Level

Vision Senor 1 X, Y (in pixels)	Vision Senor 2 X, Y (in pixels)	Time completion time (in seconds)	Attempt Status (Success/Fail)
266,361	110,111	37.59	Success
277,374	87,138	36.99	Success
338,372	343,63	39.20	Success
383,352	561,85	37.35	Fail
254,367	74,124	37.65	Success
400,367	537,120	37.58	Success
416,363	609,123	37.42	Fail
390,360	568,130	37.77	Success
407,368	593,105	37.53	Fail
249,365	65,122	37.79	Success
288,367	143,150	37.36	Success
414,371	577,142	37.93	Success

The majority of the fruitless endeavors happen when the objective article is put more remote from the mechanical arm. From figure 5 we can see that the precision is lesser for x esteems lesser than 200 or more prominent than 570. Subsequently the condition (1) does not hold useful for outrageous qualities. Successful answers for decreasing the quantity of fruitless endeavors would plan a higher request condition that can hold useful for most areas of the item and all the more adjusting by dully moving the arm until the article is in the focal point of the casing.

From the above classifications, it tends to be discovered that the time taken to get objects from the upper and lower level are nearly a similar esteem. The undertaking of grabbing an article and conveying it to the client can be finished at a normal of 37.52 seconds (37.37 seconds for the upper dimension and 37.68 for the lower level). This is essentially low contrasted with the fulfillment times determined when utilizing mind PC interface [4] or motion-based acknowledgment [5]. It can likewise be seen that endeavors are increasingly fruitful when the items are put at the upper dimension contrasted with the lower level. The explanation behind ineffective endeavours on the lower level is that the light falling on the lower level is less because of the shadow thrown by the rack. This may act like an issue for shading based article location since the situation of the identified item continues fluctuating because of deficient light.

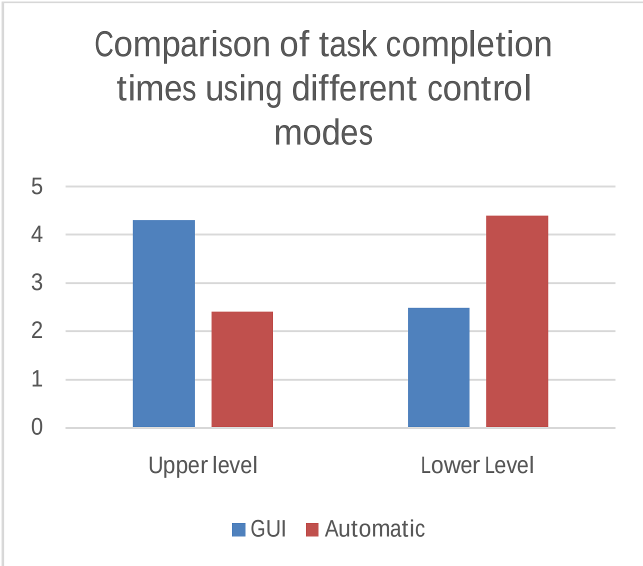

Figure 7. Comparison of task completion times using different control modes

At the point when the arm is controlled utilizing the Arm Link programming, it was seen that the consummation time for a similar errand took around 120 to 130 seconds for a talented client. Figure 7 demonstrates the correlation of the undertaking fruition times of getting the article utilizing GUI and the computerized framework. The outcome demonstrates that mechanizing the development of a robot arm can spare a great deal of time and it doesn't require the client to focus while playing out the assignment.

IV. CONCLUSION

A computer vision algorithm dependent on shading discovery has been created to mechanize the development of the automated arm. This was actualized by utilizing two vision sensors to precisely discover the area of the objective article. First sensor was utilized for acquiring the coarse area of the article and the second one was utilized for fine limitation. The mechanical arm can get objects put at various areas on the upper and lower level with a triumph rate of 83.33%. The primary objective of playing out the activity of grabbing the article under one moment is accomplished.

V. FUTURE WORK

Shading based article location was utilized to decide the situation of the ideal item. Functional applications will include objects of different shapes and numerous hues. A UI can be built up that can enable the client to choose an ideal item from the edge caught by the vision sensor by consolidating discourse acknowledgment. When the item choice is made, the directions of the article can be sustained to the arm. Profundity sensors can be utilized notwithstanding vision sensors to expand the execution of the automated arm control along these lines limiting the quantity of ineffective endeavors. Propelled vision calculations can be utilized for posture estimation and size recognition to identify same item at various edges. Fusing the systems referenced above will help with recognition of every day living articles and help wheelchair clients to perform errands in a brief timeframe.

REFERENCES

[1] “Kinova Robotics Jaco Arm”. [Online]. Available at: http://www.kinovarobotics.com/service-robotics/products/robot-arms/

[2] “Assistive Innovations Intelligent Robotic Manipulator”. [Online]. Available at: http://www.assistive-innovations.com/en/robotic-arms

[3] Camilo Perez Quintero, Oscar Ramirez, Martin Ja¨gersand “VIBI: Assistive Vision-Based Interface for Robot Manipulation”, 2015 IEEE International Conference on Robotics and Automation (ICRA)

[4] Katherine Tsui and Holly Yanco, “Development and Evaluation of a Flexible Interface for a Wheelchair Mounted Robotic Arm”, HumanRobot Interaction (HRI), 2008 3rd ACM/IEEE International Conference

[5] Indika Pathirage, Karan Khokar, Elijah Klay, Redwan Alqasemi, “A Vision based P300 Brain Computer Interface for Grasping using a Wheelchair-Mounted Robotic Arm”, 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)

[6] Hairong Jiang, Juan P. Wachs, Bradley S. Duerstock, “Integrated VisionBased Robotic Arm Interface for Operators with Upper Limb Mobility Impairments”, 2013 IEEE International Conference on Rehabilitation Robotics, Seattle, Washington USA

[7] “Robotic Arms, Trossen Robotics” ”. [Online]. Available at: http://www.trossenrobotics.com/

[8] Adrian Rosebrock, ”Ball tracking with opencv”, Sp 14, 2015 in Object Detection, Tutorials http://www.pyimagesearch.com/

[9] “Interbotix Arm Link Software, Trossen Robotics”. [Online]. Available at: http://learn.trossenrobotics.com/36-demo-code/137-interbotix-armlink-software.html