

Robotic Arm Detection in Space with Image Recognition Made in Linux with the Hough Circles Method

Roland Szabó, Aurel Gontean
Politehnica University Timișoara,
Faculty of Electronics and
Telecommunications, Applied
Electronics Department, V. Pârvan
Av., no. 2, 300223, Timișoara,
România
Email: roland.szabo@etc.upt.ro,
aurel.gontean@upt.ro

Abstract—This paper presents a method to recognize a robotic arm in space using the Hough circles method. The robotic arm has colored bottle stoppers glued at the joints, which are recognized with color filtering. After this step, the biggest colored spot is detected and marked using the Hough circle method. The joints are numbered, this way each joint's position is known. The joints can be united with lines and this way a skeleton of the robotic arm can be drawn which can be loaded in a PC to create a control application.

I. INTRODUCTION

THIS paper presents a robotic arm detection and control using the Hough circles algorithm.

The control of a robotic arm is very useful in the industry, because in today's consumer world only robotic arms can do all the tasks precisely and with high speed [1]. The robotic arms need to be automated as much as possible, because this way, they can do even more tasks with less human intervention, which will lead to higher productivity [2].

Mostly in the industry, robotic arms are blind; they don't have vision system [3]. Vision system adds artificial intelligence and more autonomy to a robotic arm [4]. Mostly in the industry robotic arms are programmed previously, they know exactly all the movements which need to be executed [14]. This way, the robotic arms are not very flexible for today's dynamic world. If the robotic arm has a vision system added, than it can change its tasks during execution, this way there is no need to stop a production line [15]. A vision system can also make auto calibration for the robotic arm, this way the robotic arm can make small adjustments during execution, this way there is no need to stop the production line and this will save money and time [16].

In this experiment a robotic arm detection and control method was made. The robotic arm has glued colored bottle stoppers, the image has applied color filters and the joints are detected with the Hough circles method [8].

The joints are numbered, so exact position of them it's known. The joints can be united with straight lines, this way the skeleton of the robotic arm can be created and this way

the movement of the robotic arm can be loaded in a PC [9]. With this information a control algorithm can be created just based on the images of the robotic arm taken with the connected cameras [10].

II. PROBLEM FORMULATION

The laboratory from the university had two Lynxmotion AL5 type robotic arms, the AL5A and the AL5B. These robots had no cameras or some other control system connected to them. The idea was to create a smart control system with image recognition, where the position in space of the robotic arm can be detected and to have a possibility to control the robotic arm using this information [11]. The intention was to implement the control system in C/C++ with the OpenCV library and run it on the Linux operating system. The used cameras were the Logitech C270 cameras which are compatible with a wide range of operating systems, including Ubuntu Linux 12.04 LTS which was used in this experiment. The Linux operating system is also a very good choice, because the system can be ported, with only a small amount of effort on other embedded devices which can run the Linux operating system.

III. PROBLEM IMPLEMENTATION

A. Used Equipment

The experiment was made with the Lynxmotion AL5B robotic arm (Fig. 1). This robotic arm is used mainly for teaching purposes, but it can be also used for tasks which reflect industrial behavior of a robotic arm, because it copies exactly the characteristics of an industrial robotic arm.

The used robotic arm is controlled with the SSC-32 servo control board (Fig. 2). The communication with the servo control board can be done from a master control system, like a PC, via the RS-232 serial interface. The SSC-32 servo control board can generate PWM signals up to 32 motors; the used robotic arm has only 6 motors, so the servo control board is more than enough. The movement angle of each motor from the robotic arm is proportional with the PWM signal's duty cycle [12].

The motors from the robotic arm are special Hitec servomotors with gears for increasing the torque; they have also a special circuit board which moves the motors proportionally to the PWM signal generated by the SSC-32 servo control board [13].



Fig. 1 Lynmotion AL5B robotic arm

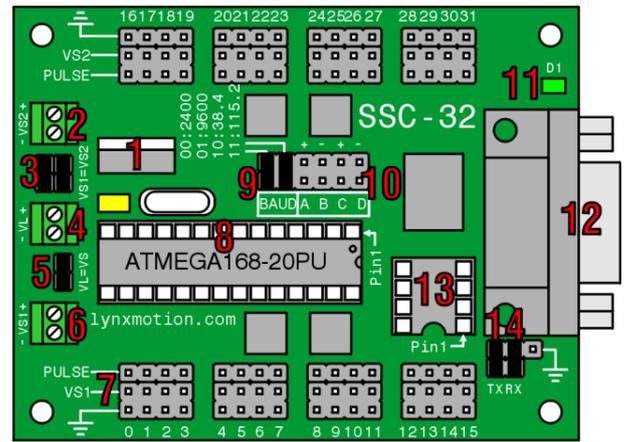


Fig. 2 SSC-32 servo control board

B. Theoretical Background

For controlling the robotic arm, SCPI (Standard Commands for Programmable Instruments) commands are sent through the serial interface from the PC to the SSC-32 servo control board. These commands were captured using the HHD Free Serial Port monitor program (Fig. 3).

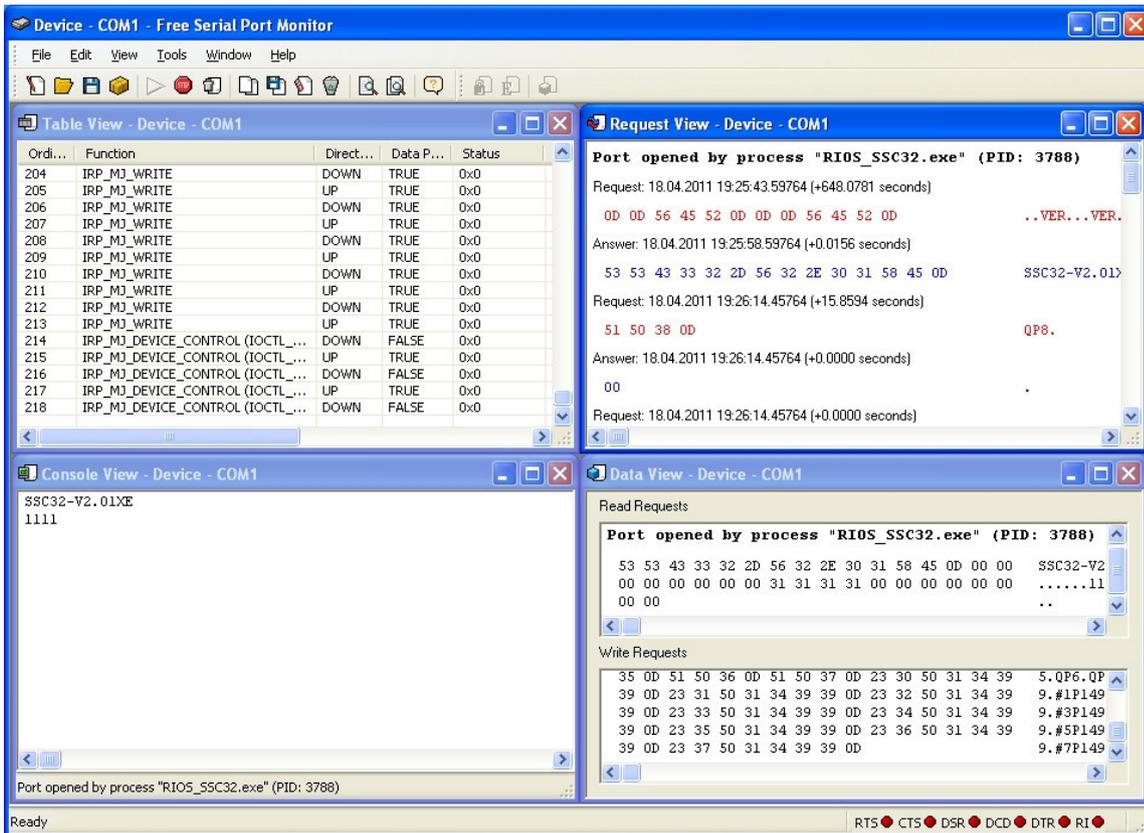


Fig. 3 HHD Free Serial Port Monitor

Next it will be presented the SCPI commands for controlling the robotic arm.

The first command is the version reading command, which is used for establishing connection over serial

interface between the master control computer and the SSC-32 servo control board.

The commands after are the commands which initialize the motors. Actually these commands test all the digital ports from the SSC-32 servo control board. There are tested

all 32 digital ports, even those which have no motors connected to them. This is a self test before controlling the robotic arm; this way is known if the system runs normally.

The “ALL SERVOS 1500” command places all servos in the position 1500. The position 1500 is the middle position of each motor. The motors are mechanically blocked to be able to make a maximum rotation of 180°, so the middle position is at 90°. The robotic arm is assembled mechanically in such a way, when all motors are in middle position, it can be said that the robotic arm woke up. The robotic arm it’s in a position which resembles a Greek capital gamma letter (“Г”). The “ALL SERVOS 1500” command places the robotic arm in the home position, a position which is well known by the system, because all the motors are at the position 1500.

The last commands are used to control each motor of the robotic arm. The first number after the “#” it’s the motor number. The second number after the “P” letter it’s the motor position. If it’s 1500, than the motor it’s in the middle position. The last number which is after the “S” letter is the execution time, which is usually set to 1000 ms. If this number is 0, than this means that the motor is set to the maximum possible speed, so this way the execution of the movement will be made at the maximum speed which can be executed by the motor.

```
// SSC-32 VERSION
\r\rVER\r

// INITIALIZE MOTORS
QPLO\rQP0\r
QP1\r
QP2\r
//...
QP31\r

// ALL SERVOS 1500 (MIDDLE POSITION)
#0P1500S0\r#1P1500S0\r#2P1500S0\r#3P1500S0\r#4P1500S0\r#5P1500S0\r
```

```
// GRIPPER
#4P1500S1000\r

// WRIST ROTATE
#5P1500S1000\r

// WRIST
#3P1500S1000\r

// ELBOW
#2P1500S1000\r

// SHOULDER
#1P1500S1000\r

// BASE
#0P1500S1000\r
```

If the minimum robotic value (500) and maximum robotic value (2500) is known, than equation (1) can be obtained.

$$\alpha = \frac{\Delta\omega}{180^\circ - 0^\circ} = \frac{2500 - 500}{180^\circ - 0^\circ} = 11, (1) \text{ robotic values} \tag{1}$$

Equation (1) implies equation (2).

$$1^\circ \sim 11, (1) \text{ robotic values} \tag{2}$$

In Fig. 4 there is the block diagram of the experiment. As it can be seen, we have a PC, the Logitech C270 webcam and the Lynxmotion AL5B robotic arm. The system is made for 2D control of the robotic arm, but for 3D control, the camera and the algorithm needs to be duplicated. The software part is a little bit more interesting. First it’s used a HSV (Hue, Saturation Value) filter. After this, the Hough circle detection it’s made.

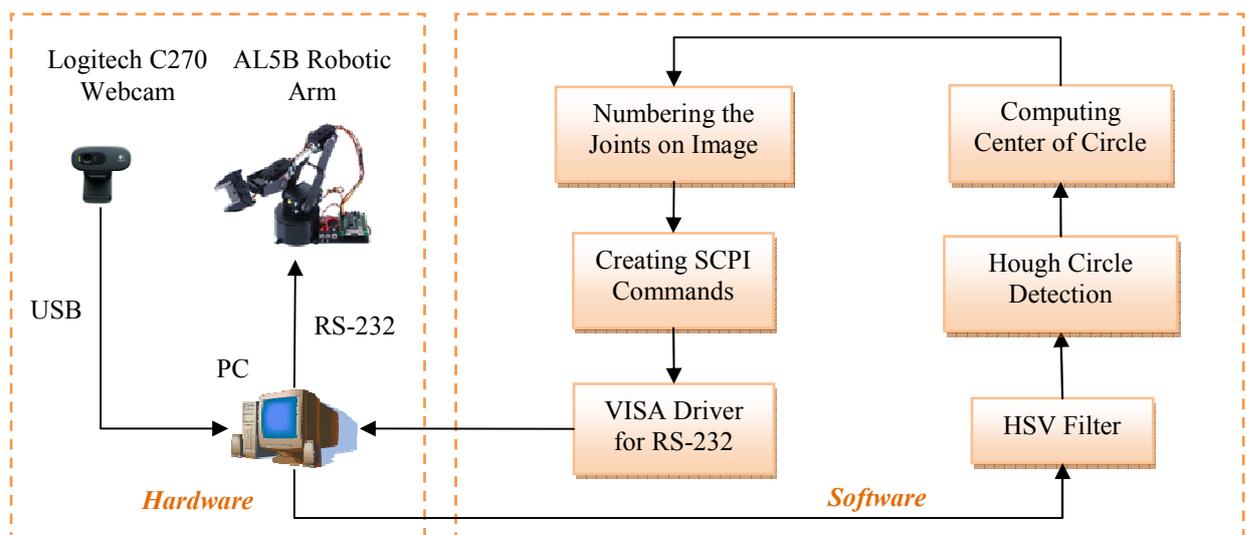


Fig. 4 Block diagram of the experiment

The Hough circle algorithm can detect round spots in an image. Round spots are present on the image after color filtering. The biggest round spot will be chosen, which will be a colored bottle stopper on a joint of the robotic arm. The Hough circle algorithm draws a circle around this detected colored spot.

After this step, the center of the circle is detected, which is actually the coordinate of a joint of the robotic arm and also is the place where the number of the joint will be placed. After this step, the joints can be united and this way the position of the robotic arm in space is known. After knowing the position of the robotic arm, SCPI commands are created and sent using the VISA RS-232 serial driver from National Instruments.

C. Software Implementation

The software was implemented in Ubuntu Linux 12.04 LTS operating system using the C/C++ programming language with the OpenCV library and the Qt GUI. As it can

be seen on the first image, the joints are recognized with the circles around the colored bottle stoppers glued to them. The joints are numbered.

The second image is the HSV filter of the original image.

The next four images are the color filters for blue, yellow, red and green with the sliders for the HSV values.

As we can see the joints were recognized, the only task is to unite the joints with straight lines, which is quite easy because the coordinates are known. They are exactly at the center of the circles. The numbers are drawn at the centers of the circles. The only task is to write the equation of a segment which goes through two points in space. After the step when the joints are united, the skeleton of the robotic arm is created. This data can be loaded in a PC and the position of the robotic arm will be known. Based on this info SCPI commands can be sent to the robotic arm to place its gripper in the desired position.

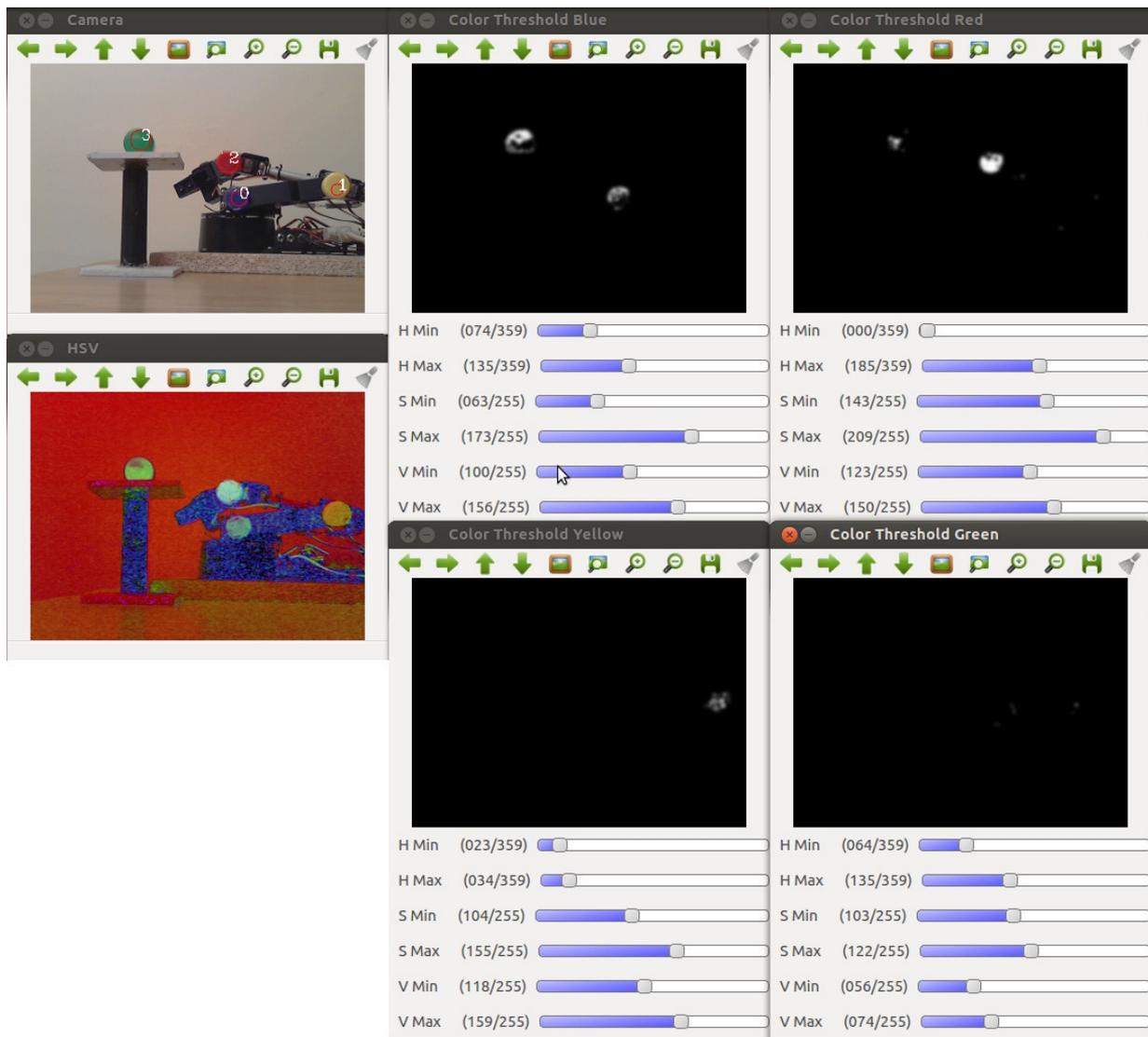


Fig. 5 Robotic arm control application under Ubuntu Linux 12.04 LTS in C/C++ programming language with OpenCV library using the Hough circles method for joint position detection

IV. CONCLUSION

It was presented a system which can recognize the joints of a robotic arm using the Hough circles algorithm.

The robotic arm recognition is useful in the industry because it can ease the maintenance of the production line. There is no need to program previously or to calibrate the robotic arm, these operations are done on-the-go during execution.

The Hough circles algorithm works flawlessly for joints detection. There was also made a test with the Hough circle algorithm to follow a bouncing ping-pong ball. The test was done with no errors. After this test it could be told that the Hough circles algorithm can follow any movement of a robotic arm, it can even follow a bouncing ping-pong ball.

The Hough circles algorithm is implemented in OpenCV, so it can be used with ease.

Further enhancements would be to port the robotic arm control application on an embedded system.

The first step would be to port it on a Raspberry PI with Raspbian Linux "Wheezy" v7.0. After this it can be ported on FPGA boards which can run Ubuntu Linux 12.04 LTS like the ZYBO or the ZedBoard development boards from Digilent.

On all the three systems the code can be ported directly as it is written in C/C++ or in Python. All the boards have the possibility to install OpenCV, so all the algorithms can be implemented with just minor changes.

The final task is to actually create the ASIC with Linux running on it and the code written in C/C++ with the OpenCV library. This would recognize the robotic arm's joints using the Hough circles algorithm. The task to create an ASIC is not the most complicated one, because there are tools from Mentor Graphics which can convert the VHDL codes in Verilog and after it can create the layout of the chip. This ASIC will behave similar to a PC or to a development board, but it will be a compact embedded IC which will be portable and will have the possibility to be placed in on PCBs with reduced size.

This idea to do everything portable and embedded came due to the tendency of the industry and consumer electronics predictions. This fact is described also in the German factories in the "Industrie 4.0" standard which explains that in the future the PC will not exist and it's no more needed. The task is to have many small smart devices like smart phones, tablets, smart watches and embedded devices which can do the same task as a PC, but they are much more flexible and much more compact. These devices are all connected in a big wireless network, all the devices communicate with each other in a smart manner and everything is more automated and works better, faster with less human intervention. This big wireless network became possible with the introduction of IP v6 which gave more IP addresses to the continuously growing devices connected to the big internet and to the intranets.

REFERENCES

- [1] W. G. Hao, Y. Y. Leck, L. C. Hun, "6-DOF PC-Based Robotic Arm (PC-ROBOARM) with efficient trajectory planning and speed control," *4th International Conference On Mechatronics*, Kuala Lumpur, 2011, pp. 1–7, <http://dx.doi.org/10.1109/ICOM.2011.5937171>.
- [2] W. Yang, J. H. Bae, Y. Oh, N. Y. Chong, B. J. You, S. R. Oh, "CPG based self-adapting multi-DOF robotic arm control," *International Conference on Intelligent Robots and Systems*, Taipei, 2010, pp. 4236–4243, <http://dx.doi.org/10.1109/IROS.2010.5651377>.
- [3] E. Oyama, T. Maeda, J. Q. Gan, E. M. Rosales, K. F. MacDorman, S. Tachi, A. Agah, "Inverse kinematics learning for robotic arms with fewer degrees of freedom by modular neural network systems," *International Conference on Intelligent Robots and Systems*, 2005, pp. 1791–1798, <http://dx.doi.org/10.1109/IROS.2005.1545084>.
- [4] N. Ahuja, U. S. Banerjee, V. A. Darbhe, T. N. Mapara, A. D. Matkar, R.K. Nirmal, S. Balagopalan, "Computer controlled robotic arm," *16th IEEE Symposium on Computer-Based Medical Systems*, New York, 2003, pp. 361–366, <http://dx.doi.org/10.1109/CBMS.2003.1212815>.
- [5] M. H. Liyanage, N. Krouglicof, R. Gosine, "Design and control of a high performance SCARA type robotic arm with rotary hydraulic actuators," *Canadian Conference on Electrical and Computer Engineering*, St. John's, CA, 2009, pp. 827–832, <http://dx.doi.org/10.1109/CCECE.2009.5090244>.
- [6] M. Mariappan, T. Ganesan, M. Iftikhar, V. Ramu, B. Khoo, "A design methodology of a flexible robotic arm vision system for OTOROB," *International Conference on Mechanical and Electrical Technology*, Singapore, 2010, pp. 161–164, <http://dx.doi.org/10.1109/ICMET.2010.5598341>.
- [7] H. Guo-Shing, C. Xi-Sheng, C. Chung-Liang, "Development of dual robotic arm system based on binocular vision," *International Automatic Control Conference*, Nantou, 2013, pp. 97–102, <http://dx.doi.org/10.1109/CACS.2013.6734114>.
- [8] R. Szabó, A. Gontean, "Controlling a Robotic Arm in the 3D Space with Stereo Vision," *21th Telecommunications Forum*, Belgrade, 2013, pp. 916–919, <http://dx.doi.org/10.1109/TELFOR.2013.6716380>.
- [9] R. Szabó, A. Gontean, "Robotic arm control in 3D space using stereo distance calculation," *International Conference on Development and Application Systems*, Suceava, 2014, pp. 50–56, <http://dx.doi.org/10.1109/DAAS.2014.6842426>.
- [10] R. Szabó, A. Gontean, "Remotely Commanding the Lynxmotion AL5 Type Robotic Arms," *21th Telecommunications Forum*, Belgrade, 2013, pp. 889–892, <http://dx.doi.org/10.1109/TELFOR.2013.6716373>.
- [11] R. Szabó, A. Gontean, "Creating a Programming Language for the AL5 Type Robotic Arms," *36th International Conference on Telecommunications and Signal Processing*, Rome, 2013, pp. 62–65. [Online]. Available: <http://dx.doi.org/10.1109/TSP.2013.6613892>.
- [12] R. Szabó, A. Gontean, "Full 3D Robotic Arm Control with Stereo Cameras Made in LabVIEW," *Federated Conference on Computer Science and Information Systems*, Kraków, 2013, pp. 37–42.
- [13] R. Szabo, A. Gontean, "Robotic Arm Control with Stereo Vision Made in LabWindows/CVI," *37th International Conference on Telecommunications and Signal Processing*, Berlin, 2014, pp. 635–639.
- [14] M. Seelinger, E. Gonzalez-Galvan, M. Robinson, S. Skaar, "Towards a robotic plasma spraying operation using vision," *IEEE Robotics & Automation Magazine*, vol. 5, issue 4, 1998, pp. 33–38, 49, <http://dx.doi.org/10.1109/100.740463>.
- [15] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," *IEEE/ASME Transactions on Mechatronics*, vol. 5, issue 1, 2000, pp. 39–48, <http://dx.doi.org/10.1109/3516.828588>.
- [16] V. Lippiello, F. Ruggiero, B. Siciliano, L. Villani, "Visual Grasp Planning for Unknown Objects Using a Multifingered Robotic Hand," *IEEE/ASME Transactions on Mechatronics*, vol. 18, issue 3, 2013, pp. 1050–1059, <http://dx.doi.org/10.1109/TMECH.2012.2195500>.
- [17] M. Kazemi, K. K. Gupta, M. Mehrandezh, "Randomized Kinodynamic Planning for Robust Visual Servoing," *IEEE Transactions on Robotics*, vol. 29, issue 5, 2013, pp. 1197–1211, <http://dx.doi.org/10.1109/TRO.2013.2264865>.
- [18] R. T. Fomena, O. Tahri, F. Chaumette, "Distance-Based and Orientation-Based Visual Servoing From Three Points," *IEEE*

- Transactions on Robotics*, vol. 27, issue 2, 2011, pp. 256–267, <http://dx.doi.org/10.1109/TRO.2011.2104431>.
- [19] N. C. Orger, T. B. Karyot, “A symmetrical robotic arm design approach with stereo-vision ability for CubeSats,” *6th International Conference on Recent Advances in Space Technologies*, Istanbul, 2013, pp. 961–965, <http://dx.doi.org/10.1109/RAST.2013.6581353>.
- [20] F. Medina, B. Nono, H. Banda, A. Rosales, “Classification of Solid Objects with Defined Shapes Using Stereoscopic Vision and a Robotic Arm,” *Andean Region International Conference*, Cuenca, 2012, pp. 226, <http://dx.doi.org/10.1109/Andescon.2012.71>.
- [21] M. Puheim, M. Bundzel, L. Madarasz, “Forward control of robotic arm using the information from stereo-vision tracking system,” *14th International Symposium on Computational Intelligence and Informatics*, Budapest, 2013, pp. 57–62, <http://dx.doi.org/10.1109/CINTI.2013.6705259>.
- [22] T. P. Cabre, M. T. Cairol, D. F. Calafell, M. T. Ribes, J. P. Roca, “Project-Based Learning Example: Controlling an Educational Robotic Arm With Computer Vision,” *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 8, issue 3, 2013, pp. 135–142, <http://dx.doi.org/10.1109/RITA.2013.2273114>.
- [23] G. S. Gupta, S. C. Mukhopadhyay, M. Finnie, “WiFi-based control of a robotic arm with remote vision,” *Instrumentation and Measurement Technology Conference*, Singapore, 2009, pp. 557–562, <http://dx.doi.org/10.1109/IMTC.2009.5168512>.
- [24] L. Haoting, W. Wei, G. Feng, L. Zhaoyang, S. Yuan, L. Zhenlin, “Development of Space Photographic Robotic Arm based on binocular vision servo,” *Sixth International Conference on Advanced Computational Intelligence*, Hangzhou, 2013, pp. 345–349, <http://dx.doi.org/10.1109/ICACI.2013.6748528>.
- [25] C. Wen-Chung, C. Chih-Wei, “Automatic Mobile Robotic Manipulation with Active Eye-to-Hand Binocular Vision,” *33rd Annual Conference of the IEEE Industrial Electronics Society*, Taipei, 2007, pp. 2944–2949, <http://dx.doi.org/10.1109/IECON.2007.4460000>.
- [26] P. C. Nunnally, J. M. Weiss, “An inexpensive robot arm for computer vision applications,” *Energy and Information Technologies in the Southeast*, Columbia, vol. 1, 10989, pp. 1–6, <http://dx.doi.org/10.1109/SECON.1989.132303>.
- [27] T. Kizaki, A. Namiki, “Two ball juggling with high-speed hand-arm and high-speed vision system,” *IEEE International Conference on Robotics and Automation*, Saint Paul, MN, 2012, pp. 1372–1377, <http://dx.doi.org/10.1109/ICRA.2012.6225090>.