

PAPER • OPEN ACCESS

Real-time vision-based grasping randomly placed object by low-cost robotic arm using surf algorithm

To cite this article: A Beyhan and N G Adar 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **938** 012008

View the [article online](#) for updates and enhancements.

You may also like

- [Facial Expression Recognition using Multiple Kinect version 2 Cameras](#)
James Purnama, Anthony Wijaya, Jayadi Taslim et al.
- [Utilising the See-and-Follow Method for Enhancing Robot Learning Ability](#)
Muh Anshar, D Halim and C Yohannes
- [Interactive Learning Media of the Solar System for Children using Kinect](#)
R F Rahmat, N Amalia, S Faza et al.



The Electrochemical Society
Advancing solid state & electrochemical science & technology

241st ECS Meeting

May 29 – June 2, 2022 Vancouver • BC • Canada

Abstract submission deadline: Dec 3, 2021

Connect. Engage. Champion. Empower. Accelerate.
We move science forward



Submit your abstract



Real-time vision-based grasping randomly placed object by low-cost robotic arm using surf algorithm

A Beyhan¹, N G Adar¹

¹ Bursa Technical University, Mechatronics Engineering Department, Bursa, Turkey

gokhan.adar@btu.edu.tr

Abstract. Vision-Based Manipulation is popular and still have open issues in robotics. The camera is a very important part of this method to obtain desired data with image processing techniques. In this study, the Dynamic position-based look and move method was selected to control the 4 DOF robotic arm. For this method, the Kinect camera was used for image processing. Kinect is a special camera which consists of both RGB and infrared camera. SURF algorithm was selected to detect a target object from the target scene using Kinect RGB camera. 3-D target object localization was calculated using Kinect infrared camera with the point cloud. The obtained target object's location is according to the camera and transformed according to the robotic arm base. Using inverse kinematics, desired joint angles were calculated according to the object position. Therefore, the robot is provided to make the desired motion and grasping the object by using gripper. Real-time implementation of the proposed method is carried out using Matlab-Simulink.

1. Introduction

Robotics is one of the most important topics of today's engineering in Industry 4.0. Because today, robots are used in a wide area from robotic arms to humanoid robots. Various cameras are used in studies on robotic arms. These cameras act as sensors for the robotic arm. The data obtained from the cameras with image processing techniques are used in the control of the robot arm. Control of the robotic arms using the data obtained from the cameras is called Visual Servoing. It is divided into four main groups [1].

Visual servo control consists of two parts: image processing and robot control. In the image processing section, the desired data is obtained from the image space. In the control part, the robot is controlled by using this data.

There are many studies on image processing techniques. The purpose of these studies is to detect the target object in the scene. Local feature detectors are used to extract the features in the image and are divided into four categories as edge, contour, corner, and region [2].

Speeded Up Robust Features (SURF) is one of the most powerful and popular image processing algorithms to detect targets [3]. It is a robust local feature detector and an improved version of the Scale Invariant Feature Transform (SIFT) algorithm [4]. There are many studies in the literature where the SIFT algorithm is used to control robotic arms [5-11].

SURF was inspired by the SIFT algorithm. Its' principles and steps are based on the SIFT. but SURF's one of the main objectives is to make computational efficiency.



Randomly placed objects were handled using a 2D vision system used for image processing while 3D cad data is used for generating 3D coordinate. The algorithm is based on the pinhole camera model. Shaver handles were selected to recognize and was implemented using MATLAB [12].

Ali et al. used an H20 mobile robot that has dual arms where each arm consists of 6 revolute joints with 6-DOF and 2-DOF grippers for grasping and placing operation for labware. The robot has a Kinect v2 sensor for image processing. SURF algorithm was used to recognize the target object. By the way forward and inverse kinematics of arm were obtained to control robot [13].

Seib [14] et al. used GiGo service robot to grasp tasks. Gigo was built with a robotic arm and a Kinect camera to recognize the target object by using the SURF algorithm.

Suligoj et al. used a multiagent robot system which is composed of two Fanuc robots. One of these robots was equipped with a stereo camera while the other was guided for tracking, tooling, or handling operations. In the image processing stage, two methods were selected to find objects which are Hough method and SURF respectively [15].

Anh et al. aimed to grasp objects using a robotic arm with an eye in hand camera. For this purpose, Image-Based Visual Servoing was used to control the robotic arm. To identify the target object SURF algorithm was selected [16].

Hofer et al. [17] proposed a Visual Servoing method to grasp objects. For this propose they used 7-DOF iARM robotic arm and RGB-D camera with a pan-tilt mechanism. In this set up to identify the object in the scenario SURF algorithm was used. Shalfan et al. [18] studied Image-Based Visual Servoing for 7-DOF robotic arm mounted on a wheelchair. SURF algorithm was selected for Visual Servoing for both eye-in-hand and eye-to-hand camera configuration. Many studies have been made about 7-DOF robotic arm with a wheelchair for disabled people [19-21].

Khusheef et al. [22] set up experiments including a 6-DOF robotic arm, low-cost camera, and conveyor belt system. An object placed on to the conveyor belt and detected using the SURF algorithm. After detecting the object, the robotic arm moved to grasp and relocate the object.

In this study, the Dynamic position-based look and move technique was selected to control the 4-DOF robotic arm with a Kinect camera.

2. Vision-based manipulation

Visual servo control is a control method that the camera uses to make the robot perform the desired motion. In this method, the camera is used as a sensor to obtain useful information for robot motion. In the literature, there are different visual servo structures. This structure consists of two main parts: Computer vision and robotic manipulation. In the computer vision part, the target object is detected in the scene using the SURF algorithm.

In the Computer vision part, the Kinect camera is used to determine its localization (x-y-z coordinate) using point cloud. In the robotic manipulation part, kinematic analysis (Forward-Inverse kinematics) is used to control and move the robot for the target object the general block diagram is given in figure 1.

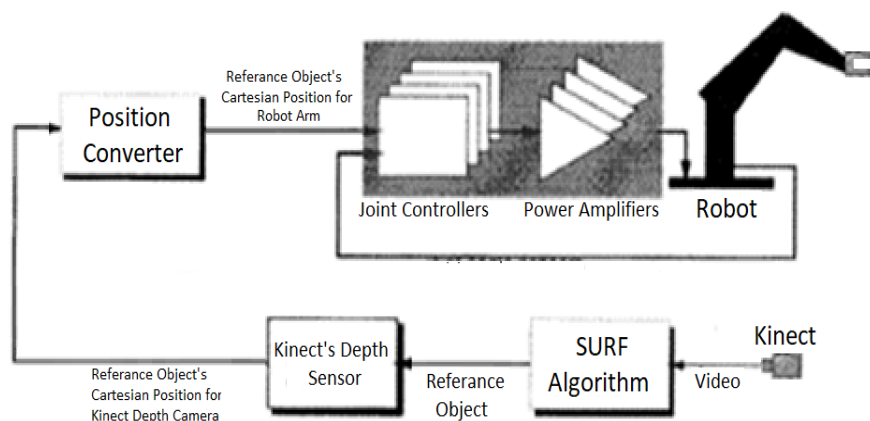


Figure 1. Vision-based manipulator general block diagram.

2.1. Vision-based object detection

2.1.1. Object detection. In the literature, different image processing algorithms for detecting the object in the image space are available. In this study, SURF which is one of the most powerful and popular image processing algorithms to detect targets algorithm was selected. It is a robust local feature detector and an improved version of the SIFT algorithm. SURF was inspired by the SIFT algorithm. Its' principles and steps are based on the SIFT but SURF's one of the main objectives is to make computational efficiency [14]. It is a scale-invariant feature detector which means that it is not affected by scaling and rotation (scale and rotation invariant detector).

To detect Interest point, SURF is used on the Hessian matrix which is as follow:

$$H = \begin{bmatrix} I_{xx}(x, \sigma_D) & I_{xy}(x, \sigma_D) \\ I_{xy}(x, \sigma_D) & I_{yy}(x, \sigma_D) \end{bmatrix} \quad (1)$$

where I_{xx} , I_{xy} , and I_{yy} are second-order Gaussian smoothed image derivatives.

In equation 1, there are second-order derivatives. These derivatives make the computation complex and take a long time. To eliminate this complexity, SURF uses of integral images method to efficiently compute the Hessian matrix.

Integral images provide a quick calculation of a given rectangular area calculating the sum of pixel values as follow:

$$I_{\Sigma}(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (2)$$

Using the integral image method, The Hessian matrix is roughly approximated.

SURF uses the Fast-Hessian detector based on the determinant of the Hessian matrix. This determinant is used to find the maximum and minimum points in the images taking the second-order derivative. The maximum determinant of this matrix represents the image regions (blob). Besides, scale and position are also found using the determinant of the Hessian matrix [7]. The determinants of the Hessian matrix are described below.

$$|H(I(x, y))| = \frac{\partial^2 I}{\partial x^2} \frac{\partial^2 I}{\partial y^2} - \left(\frac{\partial^2 I}{\partial x \partial y} \right)^2 \quad (3)$$

The determinant of Hessian can be calculated using the estimated Gaussian as follows:

$$|H_{approx}| = D_{xx}D_{yy} - 0.9D_{xy}^2 \quad (4)$$

where D_{xx} , D_{yy} , D_{xy} are the second-order Gaussian derivate which is approximated based on box filters.

2.1.2. 3-D object localization. Kinect is a special camera system developed by Microsoft (Figure 2).



Figure 2. Kinect v1 Camera.

It consists of an infrared laser emitter, an infrared camera, and an RGB camera. The RGB camera is a classic camera that is based on three basic colors (red, green, and blue). The depth detection system (3D depth sensor camera) is formed by an IR camera and infrared projector. This structure was developed by PrimeSense. The IR laser emits structured infrared lights with a certain frequency. These IR lights hit the objects and create an observed pattern onto the scene. IR camera capture this observed pattern.

Therefore, a depth map containing the distance of an object in the scene to the camera can be extracted with these two components.

2.2. Forward and inverse kinematics of robotic arm

The details of the forward and inverse kinematics of a 4-DOF robotic arm given below have been presented in [23]. Brief information was given here.

The robotic arm is given in the Cartesian coordinate system in figure 3.

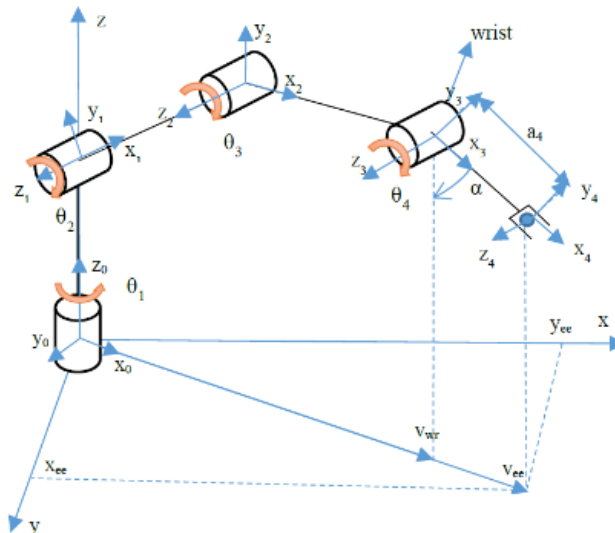


Figure 3. The coordinate frame to each link.

In figure 3 α is the wrist-roll angle, x_{ee} is x coordinate of the end effector according to the base, y_{ee} is y coordinate of the end effector according to the base, z_{ee} is z coordinate of the end effector according to the base, v_{wr} is the vector of the wrist, v_{ee} is the vector of the end effector, $\theta_1, \theta_2, \theta_3$ and θ_4 are the joint angles. Forward kinematics equations are obtained by the Denavit-Hartenberg (D-H) method. Table 1 is derived according to D-H parameters is given below.

Table 1. D-H Table of Robotic Arm.

i	a_i	α_i	d_i	θ_i
1	0	90°	45 mm	θ_1
2	150 mm	0	0	θ_2
3	150 mm	0	0	θ_3
4	90 mm	0	0	θ_4

The transformation matrix from base to end-effector (T_4^0) can be obtained as a product of matrices for every joint given in equations 5:

$$T_4^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

where P (P_x, P_y, P_z) is the position vector of the end-effector and n, o and a are orthogonal unit vectors that define the orientation of the end-effector frame.

In equation 6-8, the Position coordinates of end-effector are given below:

$$P_x = \cos(\theta_1)(a_2 \cos(\theta_2) + a_3 \cos(\theta_2 + \theta_3) + a_4 \cos(\theta_2 + \theta_3 + \theta_4)) \quad (6)$$

$$P_y = \sin(\theta_1)(a_2 \cos(\theta_2) + a_3 \cos(\theta_2 + \theta_3) + a_4 \cos(\theta_2 + \theta_3 + \theta_4)) \quad (7)$$

$$P_z = d_1 + a_2 \sin(\theta_2) + a_3 \sin(\theta_2 + \theta_3) + a_4 \sin(\theta_2 + \theta_3 + \theta_4) \quad (8)$$

In this study, the geometric approach is used to obtain the inverse kinematic equations of the robot. Using this method θ_1 , θ_2 , θ_3 , and θ_4 were calculated as follow:

$$\theta_1 = \text{atan2}(x_{ee}, y_{ee}) \quad (9)$$

$$\cos(\theta_3) = \frac{x_{wr}^2 + y_{wr}^2 - d^2 + (z_{wr} - d_1)^2 - a_2^2 - a_3^2}{2a_2 a_3} = A \quad (10)$$

$$\sin \theta_3 = \sqrt{1 - A^2} \quad (11)$$

$$\theta_3 = \text{atan2}(A, \pm \sqrt{1 - A^2}) \quad (12)$$

$$\theta_2 = \text{atan2}(x_{wr}^2 + y_{wr}^2, d_1) - \text{atan2}(a_2 + a_3 \cos(\theta_3), a_3 \sin(\theta_3)) \quad (13)$$

$$\theta_4 = \frac{\pi}{2} - \alpha + \theta_1 - \theta_2 \quad (14)$$

3. Experiment and results

To implement the Dynamic position-based look and move algorithm, the experimental setup was obtained. The experimental setup consists of a robotic arm, Kinect camera, and computer (Figure 4). Matlab-Simulink software is used to develop the code.

Robotic Arm has 5 servo motor (Dynamixel AX-12A motor). Four of them are used for joint motion while the other servo motor is used for gripper (Figure 5).



Figure 4. Experimental setup



Figure 5. 4 DOF Robotic arm

The main propose of this study is that detect the object which was selected before. To detect the object, SURF which was an image processing algorithm was used. To control the robot, the kinematics of the robot was used.

The steps of the algorithm are

- 1- Select and upload the object.
- 2- Run the SURF to extract the feature points
- 3- Find the strongest feature points
- 4- Run the Kinect camera to obtain the image scene
- 5- Match the feature points from the target object and target image scene
- 6- Find the target object
- 7- Calculate the center point of the target object
- 8- Obtain the point cloud using the Kinect 3D depth camera
- 9- Use the target object's center coordinates to calculate the position coordinates of the target point.
- 10- Transfer these coordinates to the robot base coordinates

11- Calculate joint angle using Inverse Kinematics according to these coordinates

12- Stop the algorithm until the joint angle errors below 0.01° .

For experiments, a patterned cup was selected for the target object. This object was photographed and uploaded to the computer database (Figure 6).

The first step of the SURF algorithm is to obtain feature points on the target object image in the database. The feature points obtained from the target object are given in figure 7.



Figure 6. Target object



Figure 7. The strongest feature of the target object

Different objects were placed in the target image scene together with the target object. The target image scene was obtained using the Kinect RGB camera.

The strongest feature points were found in the target image scene by SURF (Figure 8).

Feature points obtained from the target object and target image scene were compared and similar ones have been matched (Figure 9).

Using the M-estimator Sample Consensus (MSAC) algorithm, it eliminates point which does not belong to the target object (Figure 10).



Figure 8. The strongest feature of the target image

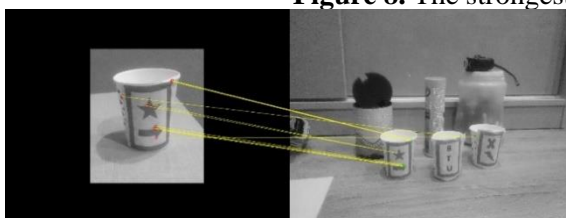


Figure 9. Matching similar points

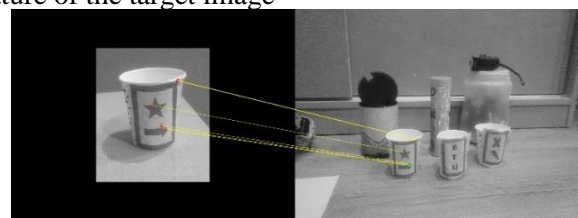


Figure 10. Elimination of unrelated point

After eliminating the unrelated point, the remaining points belong to the target object which was taken into the frame (Figure 11).

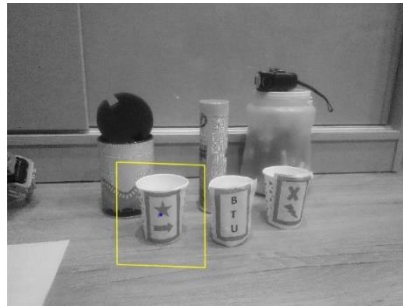


Figure 11. Assigning frame of the target object

Using this frame's corner points, the center point of the target object is calculated as follows and the center and points are showed in Figure 12:

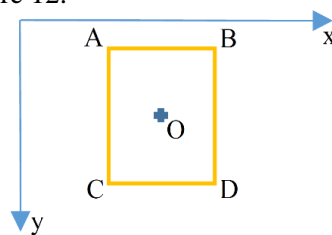


Figure 12. The center and corner point of the frame

$$O_x = A_x + \frac{D_x - A_x}{2} \quad (15)$$

$$O_y = A_y + \frac{D_y - A_y}{2} \quad (16)$$

where O_x and O_y are the center point of objects and A, B, C and D are frame's corner points.

After finding the center, the point cloud is obtained by using the Kinect 3D depth camera (Figure 13). A point cloud is a set of data points which are the x-y-z distance of the object in space. The center of the target is addressed the x-y coordinates. After having the 3D point cloud, using these addresses, the position coordinates of the object on the target scene were obtained.

The position coordinates of the target point were calculated using point cloud data with the target object's center coordinates. These target object's coordinates belong to the center of the camera. However, the position of the target object relative to the robotic arm is required. The relation between the camera center and the robotic arm's base was given in figure 14.

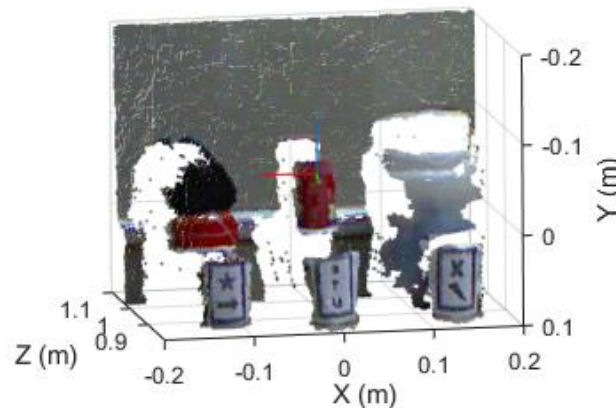


Figure 13. Point clouds of objects

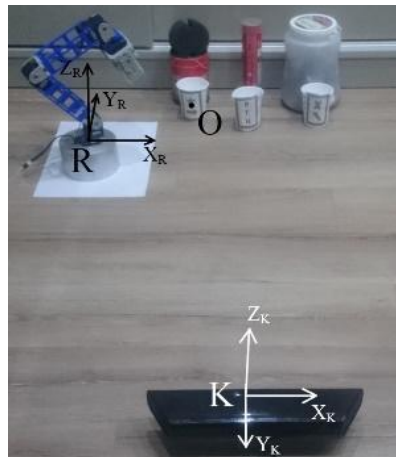


Figure 14. Image of relation Cartesian position of between Kinect and robotic arm

In figure 14, R is coordinate of the robot arm's base, K is the center coordinate of Kinect, O is the center of the object. and this position was calculated as follow:

$${}^0P_x = {}^0K_x + 25 \quad (17)$$

$${}^0P_y = {}^0K_z - 75 \quad (18)$$

$${}^0P_z = 4 - {}^0P_y \quad (19)$$

where 0P is the Cartesian position of an object according to robot arm's base and 0K is the Cartesian position of an object according to Kinect camera.

Using equation 17-19, the position of the target object relative to the robot's base has been determined. With these positions, the angle values required for the motion of the robot were calculated using Inverse Kinematics. Thus target joint angles were calculated. The robotic arm has servo motors in each joint. These servo motors have their joint angle controller. Using these angle values, the robot was moved. The robotic arm motion lasted until the angles errors were below 0.01° . After reaching the target error, the robotic arm stops to move and captured the object with its gripper (Figure 15).

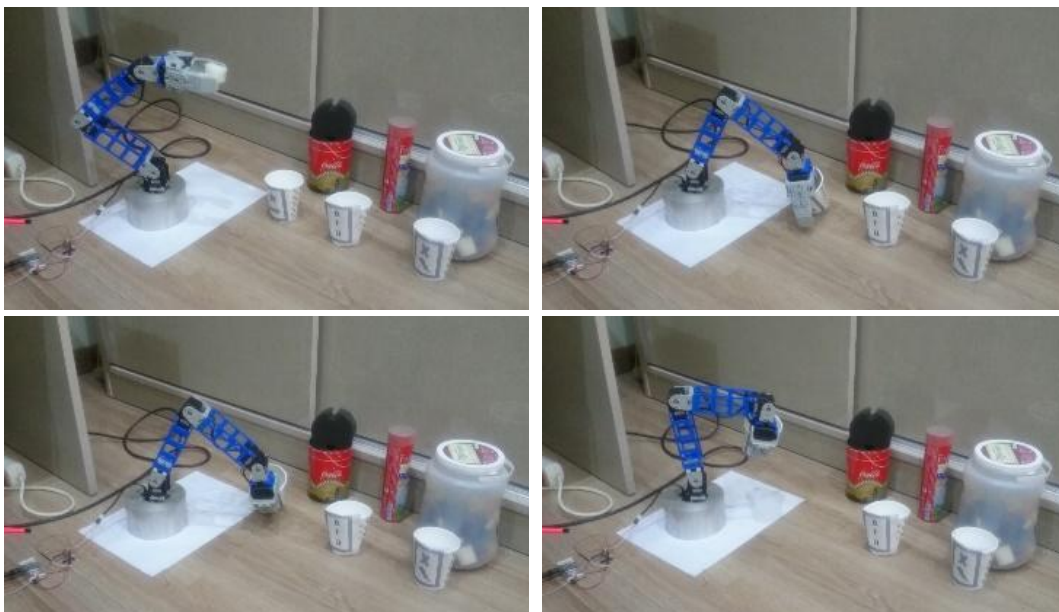


Figure 15. Move and grasp the target object

4. Conclusion

In this paper, Vision Based Robot Manipulation was implemented. 4 DOF low-cost robotic arm and Kinect camera were used for experiments. To detect the target object in the scene, the SURF algorithm was practiced with a Kinect RGB camera. The localization of the target object was found using Kinect infrared camera with the point cloud. Each joint angle was calculated using this position with Inverse kinematics. After finding the joint angles, the robotic arm was moved, and the object was grasped by the robot's gripper. The first problem of this study is the intensity of light. SURF algorithm is an image processing technique that was dependent on the intensity of light. The light intensity needs to be adjusted well to detect the target object. The second problem was the selection of the target object. The object should contain a pattern. Photographing the target object with a Kinect camera gives better results in detecting the object. The third motor of the Robotic Arm has a narrow workspace because of the mechanical limitation. Therefore, the workspace of the robotic arm has also been significantly reduced. In the future research ideas as below:

- Detecting moving objects and grasping by the gripper.
- Providing a wider workspace and designing trajectory planning for the robot.
- Assigning position of gripper according to the target object using 6 DOF robot.

Acknowledgments

This work was supported by the Scientific Research Projects Coordination Unit of Bursa Technical University. Project number: 191N008.

5. References

- [1] Hutchinson S, Hager G D and Corke P I 1996 A tutorial on visual servo control *IEEE transactions on robotics and automation* **12** 651-670
- [2] Salahat E and Qasaimeh M 2008 Recent advances in features extraction and description algorithms: A comprehensive survey *Proc. IEEE Int. Conf. Ind. Technol.* 1059-63
- [3] Bay H, Ess A, Tuytelaars T, and Van Gool L 2008 Speeded-up robust features (SURF) *Computer vision and image understanding* **110** 346-359.
- [4] Lowe D G 2004 Distinctive image features from *Int. J. Comput. Vis.* **60** 91-110
- [5] Yang Y and Cao Q 2013 A fast feature points-based object tracking method for robot grasp *International Journal of Advanced Robotic Systems* **10** 170
- [6] Sushkov O O and Sammut C 2011 Feature segmentation for object recognition using robot manipulation *Proceedings of Australasian Conference on Robotics and Automation*
- [7] Collet A, Berenson D, Srinivasa S S and Ferguson D 2009 Object recognition and full pose registration from a single image for robotic manipulation *IEEE Int. Conf. Robot. Autom.* 48-55
- [8] Chen C H and Huang H P 2013 Pose estimation for autonomous grasping with a robotic arm system *J. Chinese Inst. Eng. Trans. Chinese Inst. Eng. A/Chung-kuo K. Ch'eng Hsueh K'an*, vol. **36** 638-646
- [9] Piccinini P, Prati A and Cucchiara R 2012 Real-time object detection and localization with SIFT-based clustering *Image Vis. Comput.* **30** 573-587
- [10] Liefhebber F and Sijts J 2007 Vision-based control of the Manus using SIFT *IEEE 10th Int. Conf. Rehabil. Robot. ICORR'07* 854-861
- [11] Patil G G 2013 Vision guided pick and place robotic arm system based on SIFT *International J. Sci. Eng. Res.* **4** 242-248
- [12] Tsarouchi P, Matthaiakis S A, Michalos G, Makris S and Chryssolouris G 2016 A method for detection of randomly placed objects for robotic handling *CIRP J. Manuf. Sci. Technol.* **14** 20-27
- [13] Ali M, Liu H, Stoll N and Thurow K 2017 Grasping and placing operation for labware transportation in life science laboratories using mobile robots *Adv. Sci. Technol. Eng. Syst.* **2** 1227-37
- [14] Seib V, Thierfelder S, Paulus D. 2011 Object recognition tasks for service robots *8th Open German-Russian Workshop: Pattern Recognition and Image Understanding* 258-261

- [15] Šuligoj F, Šekoranja B, Švaco M and Jerbic B 2014 Object tracking with a multiagent robot system and a stereo vision camera *Procedia Eng.* **69** 968–973
- [16] Anh L T and Song J B 2010 Object tracking and visual servoing using features computed from local feature descriptor *ICCAS 2010 - Int. Conf. Control. Autom. Syst.* 1044–48
- [17] Hofer L, Tanaka M, Tamukoh H, Nassiraei A A and Morie T 2016 Depth-based visual servoing using low-accurate arm *2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS)* 524-531
- [18] Elarbi-Boudihir M and Al-Shalfan K A 2013 Eye-in-hand/eye-to-hand configuration for a WMRA control based on visual servoing *2013 IEEE 11th International Workshop of Electronics, Control, Measurement, Signals and their application to Mechatronics* 1-6
- [19] A. Palla, A. Frigerio, G. Meoni, and L. Fanucci, 2017 Position Based Visual Servoing control of a Wheelchair Mounter Robotic Arm using Parallel Tracking and Mapping of task objects *ICST Transactions on Ambient Systems* **4** 1–7
- [20] Jiang H, Wachs J P and Duerstock B S 2013 Integrated vision-based robotic arm interface for operators with upper limb mobility impairments *IEEE Int. Conf. Rehabil. Robot.* 1–6
- [21] Jiang H, Wachs J P and Duerstock B S 2014 Integrated vision-based system for efficient, semi-automated control of a robotic manipulator *Int. J. Intell. Comput. Cybern.* **7** 253–266
- [22] Khusheef A S, Kothapalli G and Tolouei-rad M 2011 An approach for integration of industrial robot with vision system and simulation software **5** 18–23
- [23] Beyhan A and Adar N G 2019 Detailed kinematic analysis and real-time application of the 4-dof low cost robotic arm *11th International Conference on Electrical and Electronics Engineering (ELECO)* 1037-41