BLOB Analysis for Fruit Recognition and Detection

Muhammad Dede Yusuf, RD Kusumanto, Yurni Oktarina, Tresna Dewi*, and Pola Risma

Department of Electrical Engineering, Politeknik Negeri Sriwijaya
**tresna dewi@polsri.ac.id*

ABSTRACT

Robot application in agriculture can ease the farming process, especially as the harvesting robot for seasonal fruit that is available in a short time. The addition of ”eye” as the image sensor is an important feature for a harvesting robot. Thanks to the increment of technology, the camera is getting smaller with better performance, and lower prices. The cheap sensors and components make the creation of cheap and effective robot possible. Image processing is necessary for object detection, and open source software is available now for this purpose. This paper proposes BLOB analysis for object detection of 5 fruits with different shapes and colors. The simulation results show that the proposed method is effective for object detection regardless the shapes, colors, and noises.

Keywords*:* Agriculture Robot, BLOB Analysis, Bounding Box, Logical Image

1. INTRODUCTION

Indonesia is situated in a tropical region where it enjoys abundant rain and sunshine all year long which are important factors for agriculture. The country has vast fertile soils that can grow many kinds of fruits, rice, etc. Agriculture is also still one of the main income for Indonesia and supply for the need of huge Indonesian population. However, most of the fruits are seasonal in a short time and despite a large part of Indonesia population are working in horticultural products, they are still under-performing to produce good quality. Agriculture automation can help increase the productivity and one of the automation methods is by applying robot in agriculture [1]-[13].

A robot is an automatic machine designed and created to conduct automatic task to ease human life and work, and robot applied in agriculture is called agriculture robot [14][15][17]. The main area of this robot applications is range from seeding, weed control, environmental monitoring and analysis, and harvesting.

Agriculture robot is also known as bioproduction robot [1]. The most applied type of robots for agriculture is arm-robot manipulator for its structural design is most suitable for grasping during harvesting, although another type of robots is also applicable such as a mobile robot and mobile manipulator. Agriculture robot has to equipped with sensors, and as a harvesting robot, ”eye” is a necessary feature. The increment of camera technology and at the same time camera price is also declining really helps in realizing an agriculture robot. A camera functioning as an image sensor will detect the target to be grasped [4][14][16][17][18].

The target grasping is made possible by image processing, a process to define the position of the target in image coordinate frames. One of the most applied methods in image processing for grasping an object is image segmentation. Image segmentation refers to the partition of an image into a set of a homogeneous region based on particular similarity, this process is to cluster pixels into salient image regions, i.e. corresponding to certain surfaces, objects or part of objects. Image segmentation only takes the related region and ignore the unnecessary foreground or background [2][13].

One of the methods of image segmentation for object and color detection is binary large object (BLOB) analysis. BLOB analysis has been discussed in many research in detecting the shape of an object. This paper discussed the application of blob analysis in detecting the shape and color of fruits. This paper investigates the possibility of applying this method in agriculture. Five fruits image taken with a simple webcam and simulated with an open source software SCILAB [19]. This simple and affordable method is a very important feature to realize a cheap robot for agriculture application. In order to prove the effectiveness of the proposed method, five fruits were taken within different shapes, sizes, and colors.

2. RESEARCH AND METHOD

Image segmentation used in this research is BLOB analysis which a basic technique in image processing aims to analyze the special characters in an object, therefore, this method is suitable in differentiating and extracting the shape and size of the object to be detected with its environment [20].

2.1 BLOB EXTRACTION

The objective of BLOB extracting is to isolate the BLOB (object) within its binary image. A BLOB consists of a group of connected pixel and to determine which one is neighbor pixel, and which one is not [21]. There are two most often used types of connectivity. The 8-connectivity is more accurate, however, the 4-connectivity requires fewer computation, hence faster image processing, than 8-connectivity. Figure 1 shows the difference of the 4 and 8 connectivity.

The second step is by using template matching where each feature are compared with the features of assigned type. Template matching used to find the matching area within an image with the template image or patch. By applying this method, BLOB detection can be acquired faster. The features used for template matching are dots, lines, edges, and detected area [21][22]. The steps for template matching are as follow:

1. Overlay the template on the initial image position (0,0).
2. Calculate the sum of squared differences (SSD) or the sum of absolute differences (SAD) for the overlaid area and store it in a correlation matrix.
3. Move on to the next image position and repeat step 2 until the final image position is reached.

2.2 BLOB FEATURES

After the process of BLOB extraction, the next step is to classify the different BLOBS. The first step of BLOB extraction is distinguishing each BLOB by representing it with a number of characteristics and denoted features. Feature extraction converts each

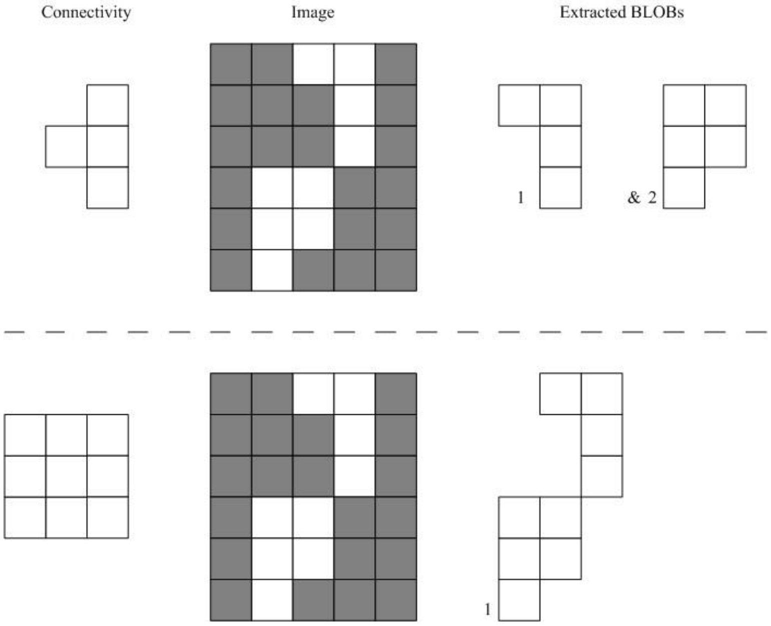


FIGURE 1. 4 and 8 connectivity

BLOB into representative numbers, in which only the relevant information is considered and ignore the rest. The first thing to do is excluding every BLOB connected to the border of an image since generally there is no information about any object outside the image. The number of a BLOB is the number of pixels within the BLOB. This feature is used to select the BLOB size, therefore, too big or too small BLOB can be ignored. Drawing a bounding circle, box, or convex hull are the way to show the detected BLOB. This paper uses the bounding circle to show the detected image.

The bounding box of a BLOB is defined as the minimum rectangle within a BLOB and used to remove the too big or too small BLOBs from an image. The rectangle is drawn by going through all the pixel for a BLOB, and finding four pixels with the minimum x value, maximum x value, minimum y value, and maximum y value. The bounding box is drawn by taking the width given by xmaxxmin and the height as ymaxymin. The bounding box is also known as the ROI (Region of Interest). Bounding box ratio of a BLOB is defined as the height of the bounding box divided by the width.

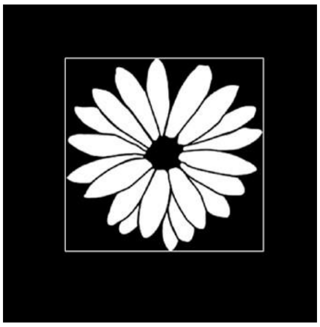


FIGURE 2. Bounding Box around BLOB

Compactness of a blob is defined as the ration of the BLOB's area to the area of the bounding circle and used as a way to distinguish the compact BLOB from the non-compact ones. The compactness of a blob is calculated by

$$\text{Compactness} = \frac{\text{Area of a BLOB}}{\text{width} \cdot \text{height}} \qquad (1)$$

Center of the mass of a binary image is the center point to balance the image, the average of x and y position of the binary object. It is defined as a point whose x value and y value are given by

$$x_c = \frac{1}{N}\sum_{i=1}^{N} x_i, \qquad y_c = \frac{1}{N}\sum_{i=1}^{N} y_i. \qquad (2)$$

Center of the bounding box is a fast approximation of the center of mass and given by

$$x_{bb} = \frac{x_{min} + x_{max}}{2}, \qquad y_{bb} = \frac{y_{min} + y_{max}}{2}. \qquad (3)$$

The perimeter of a BLOB is the length of the counter within a BLOB found by scanning along the counter of an object and summing the number of pixels encounter.

3. RESULTS AND DISCUSSION

BLOB analysis in this paper is conducted by using an open source software SCILAB that is as good as more expensive software such as MATLAB. The objects used in this paper is 5 fruits with different shape and colors. Those fruits are captured by using USB webcam and processed by Image Processing Design (IPD toolbox) or SCILAB Image and Video Processing (SIVP toolbox). The steps for image processing of BLOB analysis are as follow:

1. Capturing raw image n = camopen(1);
   RGB = avireadframe(n); figure();
   ShowColorImage(RGB, 'Original Image'); xstring(125.0,10.0,["$NAMA BUAH$"]);
   - The connected USB webcam is accessed by " n = camopen(1)".
   - USB webcam takes all the frames in term of a video, and each frame is captured and shown in a new window.
   - The captured image is named using "xtring" syntax.
2. Conducting BLOB analysis

```
Image = RGB2Gray(RGB);
InvertedImage = uint8(255 * ones(size(Image, 1),
size(Image,
2))) - Image;
Threshold = 99;
LogicalImage = SegmentByThreshold (InvertedImage,
Threshold); ObjectImage = SearchBlobs(LogicalImage);
figure();
ShowImage(LogicalImage,'Logical Image');
NumberOfObject = double(max(ObjectImage));
figure();
ShowImage(LogicalImage, 'Result of Blob Analysis',
jetcolormap(256));
```

- Converting raw image to grayscale
- Setting threshold value to 99, it is the best value for creating a grayscale image in SCILAB 5.5.2.
- The grayscale image is converted to a logical image to distinguish the desired BLOB with its environment.
- The logical image is converted to detected BLOB image, the detected image is shown with red color in a new window named "Result of BLOB analysis".

3. Drawing a bounding box

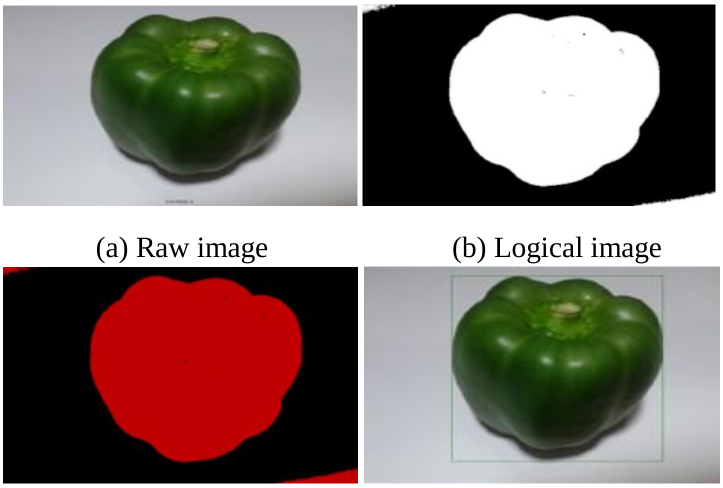
```
SizeThreshold = 1800;
FilteredObjectImage      =      FilterBySize(ObjectImage,
SizeThreshold); figure(); ShowImage(FilteredObjectImage,
'The Two Biggest Connected Areas', jetcolormap(4));
IsCalculated = CreateFeatureStruct();
IsCalculated.BoundingBox =%t;
BlobStatistics = AnalyzeBlobs(FilteredObjectImage,
IsCalculated);
FigureWindow = ShowColorImage(RGB,
        'Bounding Boxes around an Objects');
        DrawBoundingBoxes(BlobStatistics, [0 0.5 0],
        FigureWindow);
        Aviclose(1);
```

- Filtering the detected blob with a new threshold value, in this paper the value used is 1800-28000.

- Bounding box is drawn with ”DrawBoundingBoxes” function.
- RGB value for the bounding box color is given by Nilai [0 0.5 0].
- Process is completed and the camera is off by ”Aviclose(1)” function.

Five different fruits are considered in this paper with different shapes and colors, namely green bell pepper, dragon fruit, green apple, pear, and mango. These fruits are an example of unique shapes, and some come with not so clear color, but BLOB analysis algorithm can detect them.

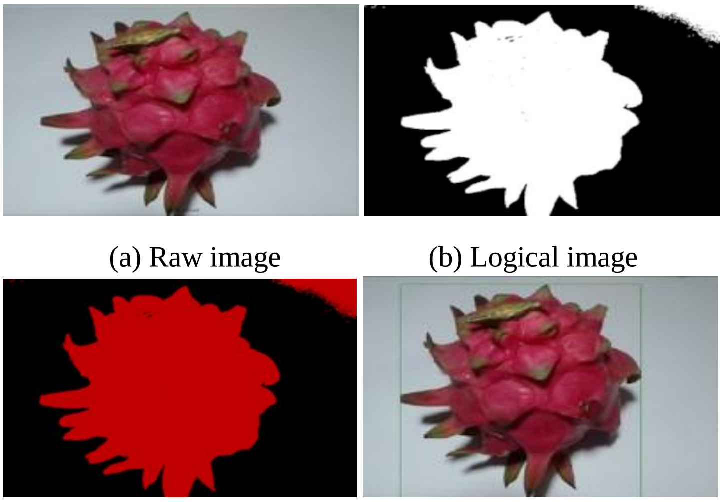
The BLOB analysis of green bell pepper (Figure 3), dragon fruit (4), green apple (5), pear (6), and mango (7) are conducted in 3 steps. The capture images by USB webcam shown in Figure 3a, 4a, 5a, 6, and 7a. The first step after converting to grayscale



(a) Raw image (b) Logical image

(c) BLOB detection (d) Bounding box

FIGURE 3. BLOB analysis of a Bell Paper

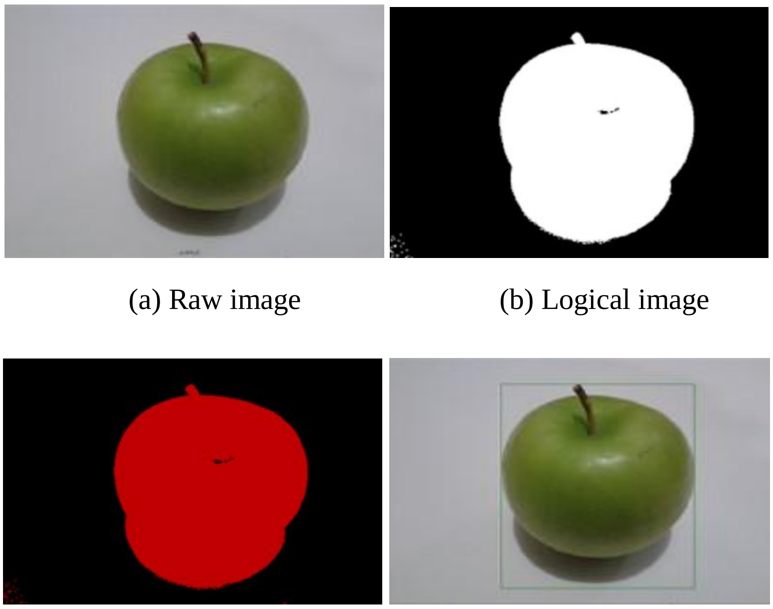


(a) Raw image (b) Logical image

(c) BLOB detection (d) Bounding box

FIGURE 4. BLOB analysis of a dragon fruit

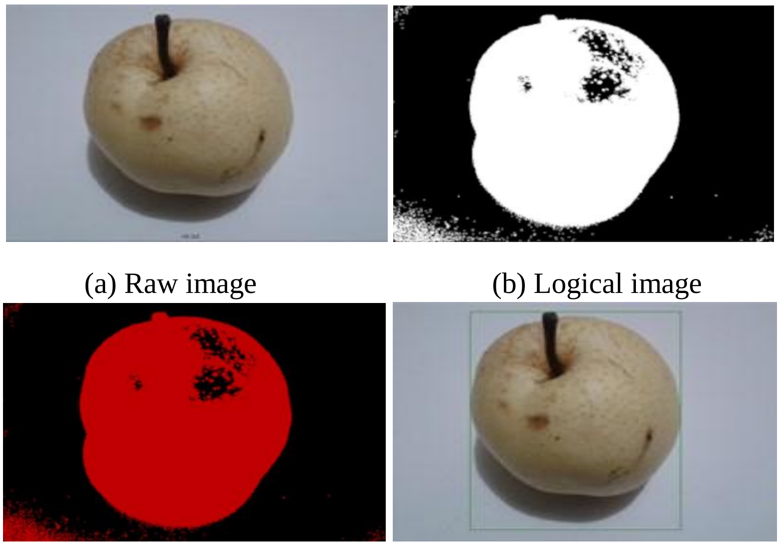
To flatten the image, logical images are acquired as shown in Figure 3b, 4b, 5b, 6b, and 7b. In logical image, the noises are still exist, therefore the resulted image detection is not smooth enough. The noises come from the occurrence of motion blurs due object motions and lighting which create color dissimilarity on object. This kind of noises are common in agriculture.



(a) Raw image (b) Logical image

(c) BLOB detection (d) Bounding box

FIGURE 5. BLOB analysis of an apple

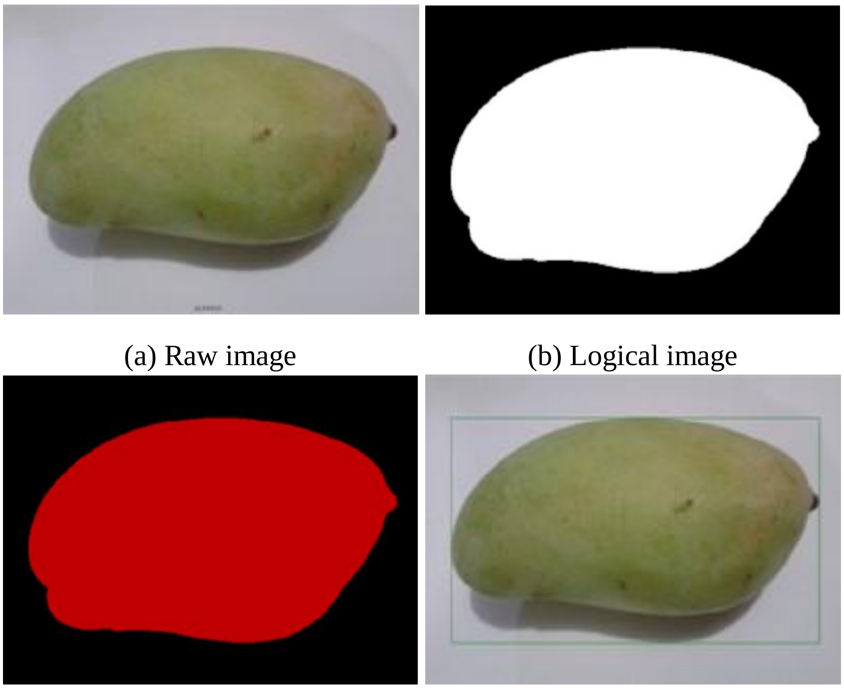


(a) Raw image (b) Logical image

(c) BLOB detection (d) Bounding box

FIGURE 6. BLOB analysis of a pear

The second process is the BLOB detection where the object is detected as shown in Figure 3c, 4c, 5c, 6c, and 7c. In BLOB detection, the objects are completely separated with their background. The third one is drawing a bounding box around the image to show the detected object as shown in Figure 3d, 4d, 5d, 6d, and 7d.



(a) Raw image (b) Logical image

(c) BLOB detection (d) Bounding box

FIGURE 7. BLOB analysis of a mango

Figure 3, 4, 5, 6, 7 show that BLOB analysis is effective for fruits detection. This method relatively cheap and easy to apply, therefore, suitable in practical application in agriculture, such as for the ”eye” of harvesting robot.

4. CONCLUSION

Robot application in agriculture can ease the farming process especially for harvesting seasonal fruit. An eye is a very crucial sensor for this kind of robot and can be provided by a simple USB webcam camera. Image processing for object/target detection can utilize an open source software such as SCILAB. This paper presents BLOB analysis for fruit detection, and 5 fruits with different shapes and colors are used as the examples. Simulation results show that BLOB analysis is successfully applied to all type of fruits. This method can overcome noises that might occur in agriculture such as motion blur and lighting effects. Therefore, the proposed method is applicable for a harvesting robot.

REFERENCES

[1] M. Monta, “Robots for Bioproduction Systems”, IFACProceedingsVolumes, Vol 33, No. 29, pp. 1-11, 2000. https://doi.org/10.1016/S1474-6670(17)367447.

[2] A. Bala, “An Improved Watershed Image Segmentation Technique Using MATLAB,” Int.J.Sci.Eng.Res, vol. 3, no. 6, p. 1206, 2012.

[3] A. Silwal, A. Gongal, and M. Karkee, “Identification of red apples in field environment with over the row machine vision system, vol. 16, no. 4, pp. 66-75, 2014.

[4] R. Barth, J. Hemming, and E. J. van Henten, “Design of an Eye-inHand Sensing and Servo Control Framework for Harvesting Robotics in Dense Vegetation”, Biosystems Engineering, Vol. 146, pp. 7184, 2016. http://dx.doi.org/10.1016/j.biosystemseng.2015.12.001.

[5] W. Ji et al. 2012 “Automatic Recognition Vision System Guided for Apple Harvesting Robot Computers and Electrical Engineering”, Computers & Electrical Engineering, Vol. 38, No. 5, pp. 11861195. https://doi.org/10.1016/j.compeleceng.2011.11.005

[6] M. Stoelen, K. Krzysztof, V. F. Tejada, N. Heiberg, C. Balaguer, and A. Korsaeth, “Low-Cost Robotics for Horticulture: A Case Study on Automated Sugar Pea Harvesting”, 10th European Conference on Precision Agriculture (ECPA), 2015. DOI: 10.3920/978-90-8686-814-8 34.

[7] T. T. Nguyen, E. Kayacan, J. De Baedemaeker, W. Saeys, “Task and Motion Planning for Apple Harvesting Robot”, IFAC Proceedings Volumes, Vol. 46, Issue. 18, pp. 247-252, 2013. https://doi.org/10.3182/20130828-2-SF-3019.00063.

[8] G. Wang, Y. Yu, and Q. Feng, “Design of End-Effector for Tomato Robotic Harvesting”, IFAC-PapersOnLine, Vo. 49, Issue 16, pp. 190-193, 2016. https://doi.org/10.1016/j.ifacol.2016.10.035

[9] J. Li, M. Karkee, Q. Zhang, K. Xiao, and T. Feng,“Characterizing Apple Picking Patterns for Robotic Harvesting”, Computers and Electronics in Agriculture, Vol. 127, pp.633640, 2016. http://dx.doi.org/10.1016/j.compag.2016.07.024.

[10] G. Feng, C. Qixin, and M. Nagata, “Fruit Detachment and Classification Method for Strawberry Harvesting Robot”, International Journal of Advanced Robotic Systems, Vol. 5, Issue 1, pp. 4148, 2008. https://doi.org/10.5772/5662.

[11] S. Amatya, M. Karkee, Q. Zhang, and M. D. Whiting, “Automated Detection of Branch Shaking Locations for Robotic Cherry Harvesting Using Machine Vision”, Robotics, Vol. 6, Issue. 4. Pages 31, 2017 https://doi.org/10.3390/robotics6040031.

[12] S.S. Mehta and T. F. Burks, “Robust Visual Servo Control in the Presence of Fruit Motion for Robotic Citrus Harvesting”, Computers and Electronics in Agriculture, Vol. 123, pp. 362375, 2016. http://dx.doi.org/10.1016/j.compag.2016.03.007.

[13] C. Wang, Y. Tang, X. Zou, W. SiTu, and W. Feng, “A Robust Fruit Image Segmentation Algorithm against Varying Illumination for Vision System of Fruit Harvesting Robot”, Optik, Vol. 131, pp. 626-631, 2017. http://dx.doi.org/10.1016/j.ijleo.2016.11.177.

[14] H. Wang and Y. Liu, “Dynamic Visual Servoing with an Uncalibrated Eye-inHand Camera”, Robot Manipulators, IntechOpen, ed. Marco Ceccarelli, 2008. DOI: 10.5772/6222.

[15] T. Dewi, P. Risma, Y. Oktarina, and M. T. Roseno, “Perbandingan Desain dan Pemodelan Tangan Robot Lima Jari sebagai Sistem Underactuated”, AnnualResearchSeminar2016, Vol 2, No. 1, pp. 139-145, 2016.

[16] E. Bugarin and R. Kelly, “Direct Visual Servoing of Planar Manipulators Using Moments of Planar Targets”, Robot Vision, IntechOpen, ed. Ales Ude, 2010. DOI: 10.5772/9297.

[17] M. F. Reis, A. C. Leite, P. J. From, L. Hsu, and F. Lizzaralde, “Visual Servoing for Object Manipulation with a Multifingered Robot Hand”, IFAC-PapersOnLine, Vol. 48, issue. 19, pp. 1-6, 2015. https://doi.org/10.1016/j.ifacol.2015.12.001

[18] N. T. Siebel, D. Peters and G. Sommer, “Models and Control Strategies for Visual Servoing”, Visual Servoing Rong-Fong Fung, IntechOpen, 2010, DOI: 10.5772/8550. https://www.scilab.org/, accessed on June 15th, 2018.

[19] J. F. Khan, S. M. A. Bhuiyan, and R. R. Adhami, “Image segmentation and shape analysis for road-sign detection,” IEEETrans.Intell.Transp.Syst, vol. 12, no. 1, pp. 83-96, 2011.

[20] A. Kaspers, “Blob Detection.

[21] M. Dhome, U. Blaise, and P. C. Ii, “Real Time Robust Template Matching,”, 2003.