[[1]](#footnote-1)

Template for Preparation of Papers for IEEE Sponsored Conferences & Symposia\*

A. Loh, S. Hu, Department of Computer Science and Information Engineering, National Taiwan University

*Abstract*— Through a combination of contour detection, color recognition, and length measurement, it is possible to program a robotic arm to autonomously detect and stack blocks into a stable structure.

# INTRODUCTION

One of the common applications of robotic arms is the assembly of parts. However, rather than manually control the arm to do this, it would be a much more efficient use of manpower to program a level of autonomy into the arm to handle certain tasks by itself. One way to do this would be to use a camera to first detect the location of objects, and then feed the coordinates to the robotic arm to process. In addition, by finding the perimeter of each detected object and also doing color segmentation in the captured images, a level of object discernment can be achieved that will allow for some manner of flexibility in how the robotic arm can differentiate each object.

# Related Work

# Method

The methodology employed in this project can be divided into several parts: camera calibration, image capture and contour detection, color segmentation, centroid and principal angle calculation, coordinate conversion, and robotic arm control.

## A. Camera Calibration

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

## B. Color Segmentation

Color segmentation is done through the HSV color spectrum. Images captured by the camera are first converted from RGB format to HSV. The images are then put through a threshold, where the saturation and value/lightness value ranges are adjusted to reduce noise and show the target objects in as clean a manner as possible. The color segmentation part comes in when setting the hue value range to single out the four sets of colored blocks that were being used: blue, red, yellow, green. A little research was done to find that the hue values of the four basic colors were 75-130 for blue, 160-179 for red, 22-38 for yellow, and 38-75 for green. While similar, due to the lighting conditions in the setting used for the experiment, the actual value ranges used were a little different.

After color segmentation, the resulting image would show white on areas of the segmented color and black everywhere else, making contour detection easier to do.

## C. Contour Detection

Contour detection is largely done through existing OpenCV’s library of functions. In particular, Canny is used to detect edges within the image, which is then processed through findContours() to get the actual contours.

To explain Canny without going into too much detail, according to OpenCV’s documentation, noise is first filtered from the image using a Gaussian filter. An intensity gradient is then made from the image, where non-maximum suppression is applied to remove pixels that are not considered to be part of an edge. Finally, an upper and lower threshold is applied on the resulting pixels, where if the pixel is below the lower threshold or not connected to a pixel above the upper threshold, it gets rejected.

findContours() takes the results and finds any contours that can be drawn among the remaining pixels. This is done, according to OpenCV’s documentation, through a border following process written by Suzuki, S. and Abe, K.

## D. Calculations

For each contour, its moment is calculated through OpenCV’s built-in function moments(), which uses Green’s theorem’s formula for doing so. By finding the moments of each contour, the centroid can also be calculated by finding the x-coordinate by dividing the contour’s moment at the coordinates (1, 0) with the moment at (0, 0), and the y-coordinate by dividing the moment at (0, 1) with (0, 0).

Having found the centroid of each object within camera view, every object outside of the robotic arm’s field of operation gets filtered out by defining the coordinates of the four corners of the field and ignoring everything that lies outside of it.

The coordinates that get fed to the robotic arm is determined by manually positioning the robotic arm above the center of the object as seen in Figure 1. The x and y coordinates of the gripper’s position are then retrieved from the software controlling the robotic arm and recorded as training points. These training points are used to find the correlation between the camera coordinates and the real world coordinates used by the arm, and are recorded into an Excel sheet to calculate the line of best for x and y, as seen in Figure 2. The resulting equations for the x and y coordinates are then applied directly into the program code itself to handle the conversion from centroids of detected objects on camera to actual coordinates that are fed into the robotic arm to maneuver to. The z coordinate is found by simply manually recording the values used when positioning the robotic arm a certain distance above the table top used in the experiment.

The principal angle is found by performing PCA analysis on the original image and the set of contour points to find the eigenvectors of each contours, which is then plugged into the atan2() function to find the resulting angle.

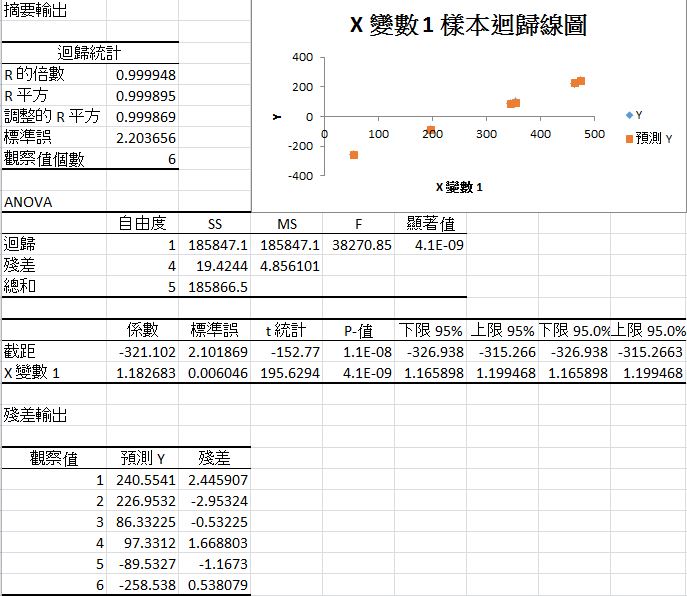


Figure 2



Figure 1

## E. Robotic Arm Control

# Experiment And Results

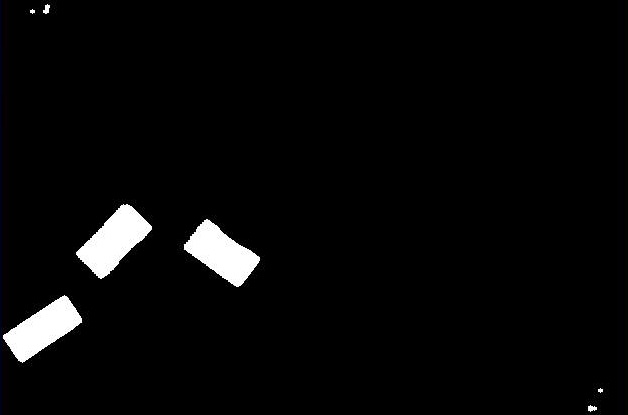
Overall, with a few shortcomings and limitations, the robotic arm was able to accomplish the task of stacking a stable structure autonomously.

## A. Color Segmentation

The main crux of the problem lied in detecting the colored blocks after color segmentation. With the camera image Figure 3, it was found that a saturation value range of 1 to 255, value/lightness range of 100 to 255, and a x and y border from 0 to 600 and 500, respectively, produced the most optimal image in color and object detection, as seen in Figure 4.

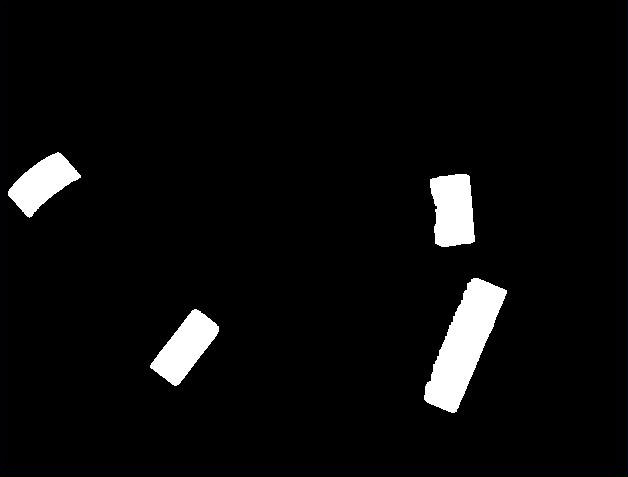
## C:\Users\Loh-Shilin\Desktop\cap8.jpgHowever, when it came to setting the hue ranges, it was initially found that the lighting of the room made it difficult to detect certain colors due to the light’s reflection. This was compensated by using a plastic mat to reduce the glare casted on the area picked up by the camera.

Since only the colors yellow, green, blue, and red were used, the hue value ranges were set to be 10 to 40, 44 to 95, 95 to 114, and 140 to 179, respectively for each color. The resulting color segmented images can be seen in Figures 5 through 8.

Even then, there were still problems detecting the color red, as seen in the imperfect contour drawn for the red triangle in Figure 3 and 4. The problem seemed to be that since red lies at the extreme end of the color spectrum, the hue value range for it actually extends past 179 and overlaps back to 0, seeing how the camera was able to perfectly pick up red blocks when viewed under the full 0 to 179 spectrum. Limitations in OpenCV’s coding library and time constraint prevented a solution to be made to include two ranges of hue values to be used during the segmentation process, and since a specific shade of red could still be detected, only blocks of that shade were used for red blocks to help alleviate the problem. 

## B. Identify the Headings

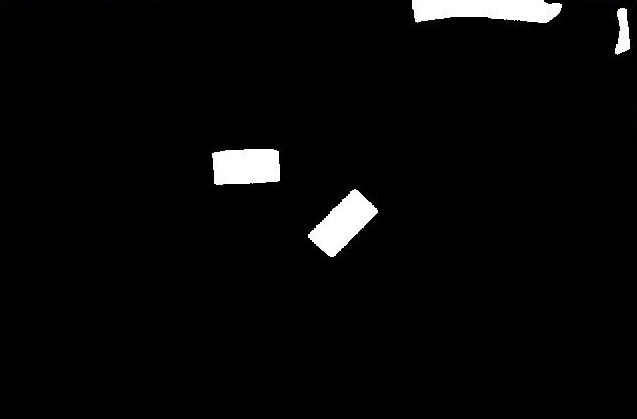
Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## C. Figures and Tables

### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

a. Sample of a Table footnote. (Table footnote)

1. Example of a figure caption. *(figure caption)*

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K.”

# Conclusion

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

Appendix

Appendixes should appear before the acknowledgment.

Acknowledgment

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression, “One of us (R. B. G.) thanks . . .” Instead, try “R. B. G. thanks”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

References

1. G. O. Young, “Synthetic structure of industrial plastics (Book style with paper title and editor),” in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
2. W.-K. Chen, *Linear Networks and Systems* (Book style)*.* Belmont, CA: Wadsworth, 1993, pp. 123–135.
3. H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.
4. B. Smith, “An approach to graphs of linear forms (Unpublished work style),” unpublished.
5. E. H. Miller, “A note on reflector arrays (Periodical style—Accepted for publication),” *IEEE Trans. Antennas Propagat.*, to be published.
6. J. Wang, “Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication),” *IEEE J. Quantum Electron.*, submitted for publication.
7. C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.
8. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style),” *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9th Annu. Conf. Magnetics* Japan, 1982, p. 301].

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord “Format” pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.

1. M. Young, *The Techincal Writers Handbook.* Mill Valley, CA: University Science, 1989.
2. J. U. Duncombe, “Infrared navigation—Part I: An assessment of feasibility (Periodical style),” *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34–39, Jan. 1959.
3. S. Chen, B. Mulgrew, and P. M. Grant, “A clustering technique for digital communications channel equalization using radial basis function networks,” *IEEE Trans. Neural Networks*, vol. 4, pp. 570–578, July 1993.
4. R. W. Lucky, “Automatic equalization for digital communication,” *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.
5. S. P. Bingulac, “On the compatibility of adaptive controllers (Published Conference Proceedings style),” in *Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory*, New York, 1994, pp. 8–16.
6. G. R. Faulhaber, “Design of service systems with priority reservation,” in *Conf. Rec. 1995 IEEE Int. Conf. Communications,* pp. 3–8.
7. W. D. Doyle, “Magnetization reversal in films with biaxial anisotropy,” in *1987 Proc. INTERMAG Conf.*, pp. 2.2-1–2.2-6.
8. G. W. Juette and L. E. Zeffanella, “Radio noise currents n short sections on bundle conductors (Presented Conference Paper style),” presented at the IEEE Summer power Meeting, Dallas, TX, June 22–27, 1990, Paper 90 SM 690-0 PWRS.
9. J. G. Kreifeldt, “An analysis of surface-detected EMG as an amplitude-modulated noise,” presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.
10. J. Williams, “Narrow-band analyzer (Thesis or Dissertation style),” Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.
11. N. Kawasaki, “Parametric study of thermal and chemical nonequilibrium nozzle flow,” M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.
12. J. P. Wilkinson, “Nonlinear resonant circuit devices (Patent style),” U.S. Patent 3 624 12, July 16, 1990.

1. [↑](#footnote-ref-1)