

Sticker Detection and Counting for Voting Boards

Giyeon Kim, Ilhyun Jo, Seongwon Bang

Seoul National University

Seoul 151-742, Republic of Korea

{gompul23, onemonster, makesource}@snu.ac.kr

Abstract

An application for regional sticker counting for voting boards. Counting stickers on voting boards was implemented using circular Hough transform and template matching. Color based categorization was done by k-mean clustering.

1. Introduction

For our project, we want to create an application that uses regional partitioning and object recognition/counting to automate the counting process for voting boards. There are many cases when people need statistics to back their research and if there isn't enough data online, people will often go out to the streets and gather the data themselves. When doing so they often use voting boards with a set of colorful stickers. They politely ask the people on the streets to express their opinion by placing one of those stickers on the section of the board they are inclined to. After a while the board will be filled with stickers and the researchers can go back home and count the votes. We thought creating an application that counts the votes automatically will have practical value.

In small projects with one voting board and tens of votes, this counting process might not be such a big deal, but what if there were several voting boards each with hundreds of votes? The counting process will take a very long time with a large error margin. We thought that if we use image recognition to count the stickers, we could automate this counting process without worrying about human error. So the application that we needed to develop was one that can take a picture of a voting board as input then output the number of stickers per color in a certain region.

2. Background

2.1. CHT (circular hough transform)

Circular hough transform is a type of hough transform method that's designed to find circles in a digital image. The CHT algorithm for short, sends points in the coordinate space to the hough space where each point becomes a circle. If circles overlap enough times on a single point in the hough space, that point is used to detect a circular object. For example, let's assume there is a circle in the coordinate space where the center is at (a, b) and the radius is r . This circle can be represented with the equation

$$(x - a)^2 + (y - b)^2 = r^2$$

A point on that circle (x_1, y_1) , can give us the equation

$$(x_1 - a)^2 + (y_1 - b)^2 = r^2$$

If you transform this equation, you can alter it to be represented in the hough space where the x and y axis becomes the a and b axis, creating a circle whose center is x_1, y_1 and the radius is r . Therefore, every point in the coordinate space can be represented as a circle with radius r in the hough space.

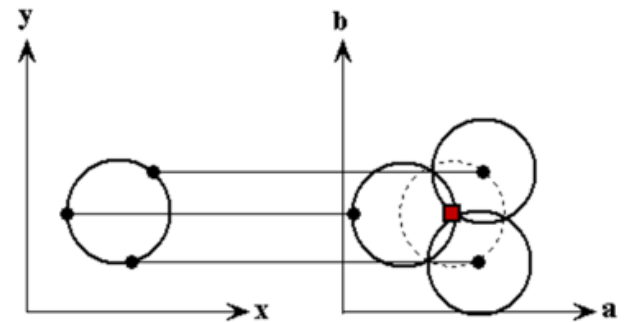


Figure 1. The Circular Hough Transform

In this way, by sending every point in the coordinate space to the hough space, the circles drawn on the hough space can vote for which (a, b) value should be chosen as the center of the circle in the coordinate space. Since it is a voting based algorithm, selecting an appropriate threshold value can show good results even for cases where there is lots of noise or the circle is distorted.

2.2. Cross correlation

Cross correlation is a value used to measure the similarities between signals. It is often used for pattern recognition, single particle analysis, and electron tomography. The definition for cross correlation is the following equation below. For image f , and the template t to compare f with, the cross-correlation value is represented as the sum of the pixel-wise multiplication between the two.

$$R_{tf}(i, j) = \sum_m \sum_n f(m, n) t(m - i, n - j)$$

If the image and template are identical the cross correlation becomes very large and if they are not identical, the value would be relatively much smaller. Therefore, this value can be used as the standard for whether the image fits the template.

2.3. K-means clustering

K-means clustering is an algorithm that categorized a set of data into k different categories based on a given standard. The objective of this algorithm is to divide data set of size n into k different clusters where the variation of distance between data within each cluster is the minimum. This algorithm works in the following process. First the algorithm separates the data into k different clusters randomly. Then the centroid of the cluster is calculated and each data piece gets reassigned to the centroid closest to it. This process of calculating the centroid and reassigning is repeated until all the data stops getting reassigned to different centroids.

3. Related work

There is already lots of research done on the topic of template matching. Out of them, references [1], [2], and [3] are papers on circle detection that managed to get a high detection rate. [1] and [2] especially, are the research papers used to implement `imfindcircle` in matlab and you can check the efficiency and accuracy of this function in these papers. [4] and [5] are about template matching and shows several different attempts to get a better matching accuracy

more efficiently. In paper [4], they used a matching filter to get better results in rotation invariant matching.

4. Methodology

Most voting boards have different regions that people can vote for. Therefore, we let the user decide the region in which the stickers should be counted. We also let the user set the template for the sticker that the user needs counting.

4.1. CHT approach

Using the size of the template from the user input, we set the value for the circle's radius. Then we used this radius value with the matlab's built-in function `imfindcircles`, which is implemented using circular hough transform, and as a result, got the locations that were thought to be the center of circles. With the appropriate threshold value for this function, this function was capable of detecting stickers that overlapped or was halfway sticking out from the region. After finding the stickers, we used k-mean clustering to categorize the stickers that were detected by colors, and the result was shown in a chart.

4.2. Cross correlation approach

Since one of the important factors of sticker detection (like most algorithms in general) is speed of the execution, we thought that the resolution of the stickers was not a big issue. So to speed up the sticker counting process we resized the image to be smaller before applying cross correlation. After resizing, we calculated the cross correlation between the input template and region given using the `xcorr2` function. To get a cross correlation value invariant to the orientation of the sticker, we rotated the template and used the biggest value as the final correlation value for each portion of the region

$$result(k, l) = \max_{1 \leq i \leq n} C_i(k, l)$$

In the equation above, $C_i(k, l)$ is the cross correlation value when the template is rotated by the i th angle. With the value calculated with the cross correlation equation altered as such, and using non maximal suppression, we detected the regions with the highest correlations to the template. Then we used k-mean clustering the same way we did in the first approach and got a chart with the sticker count for each color.

5. Experiments

5.1. Circular stickers

We tested CHT and cross correlation on the same voting board. Figure 1. (a) is the voting board images used as input.

Figure 1. (b) shows the flow chart and result image for each step when using the CHT method. The edge and sensitivity threshold was set to 0 and 0.925 respectively. Figure 1. (c) is the flow chart and result image for each step for the cross correlation method. We used the Sobel edge filter with edge threshold 0.015. The correlation value's lower bound was 0.35. Since the images resolution was already low, the resize coefficient was kept as 1.



Figure 2. (a) Voting Board Image

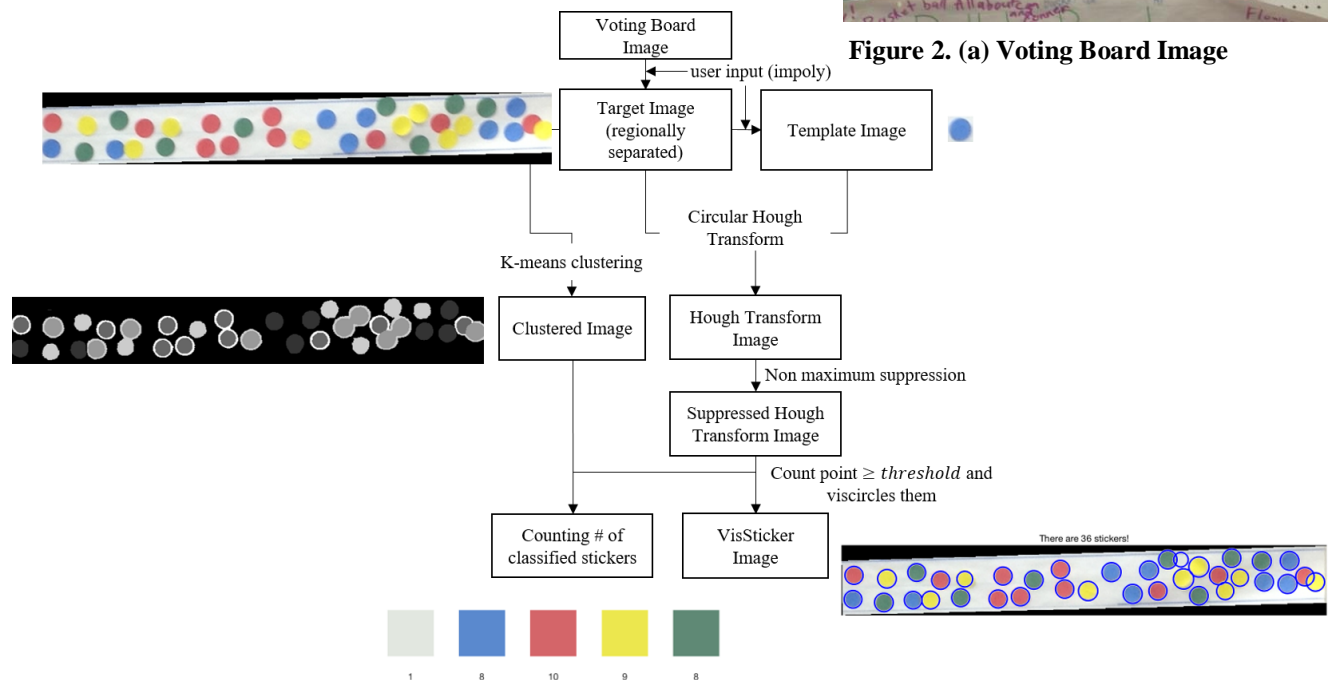


Figure 2. (b) The Flow Chart of CHT method

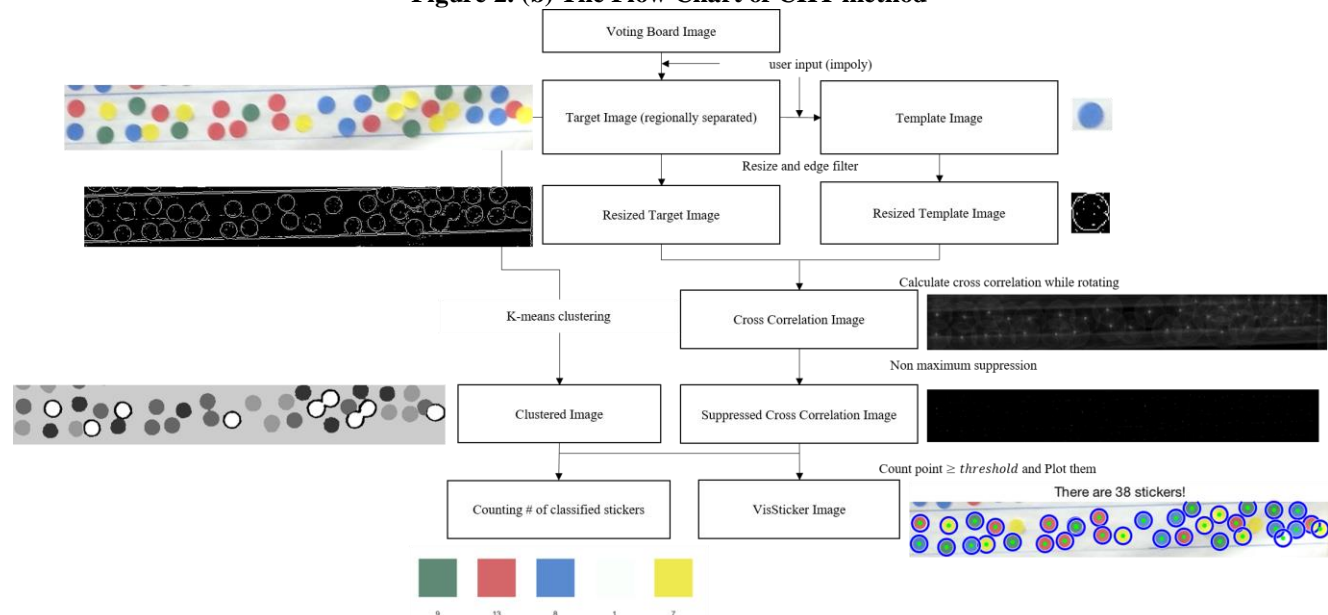


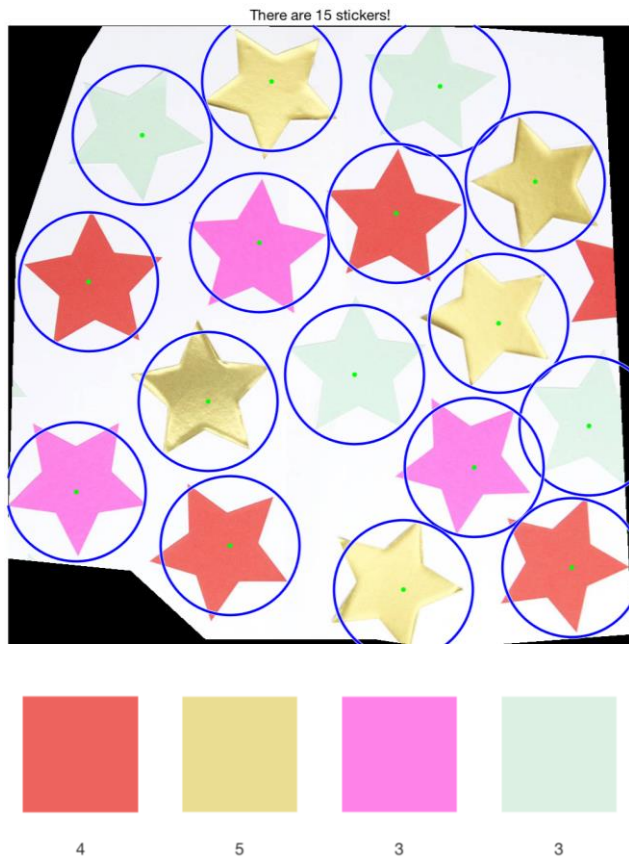
Figure 2. (c) The Flow Chart of Cross Correlation Template Matching

Table 1. The Result of CHT and Cross Correlation

method	# of stickers	False positive	False negative	accuracy
Original	35	-	-	-
CHT	36	1	0	0.97
CCTM	38	5	2	0.87

5.2. Noncircular stickers

The second method uses template matching, therefore it is possible to count the number of sticker without shapes. We used star shape stickers to check how well it works. In this experiment, we used canny edge filter with resize factor was 0.5 and correlation low bound was 0.65. The result is shown in Figure 2.

**Figure 3. The Result of Detecting Star Shape Stickers**

6. Conclusion

6.1. Discussion

CHT approach:

For cases where the sticker is a circle, the results were very accurate finding even squashed or partially cut off circles. The running time was very fast without losing

accuracy. The downside of this solution was that this solution only works for circle stickers. However, since most voting board stickers are circular, this solution seemed viable and quite useful for most use cases.

Template matching approach:

The advantages of using this approach was that it would work even if the stickers aren't circular. Since the template rotates, template matching will work regardless of the stickers orientation. Similarly, to the first approach, by finding the appropriate threshold for the correlation value, finding squashed or cut off stickers was also possible.

Unfortunately, for both approaches there was a problem when it came to detecting yellow circles. The problem was unrelated to the imfindcircles or template matching functions itself and was that the yellow stickers were too similar to the white background and it was hard to find the edges of the stickers. To solve this problem, we tried to lower the threshold value but as we lowered the threshold value the noise made it so that we couldn't get the correct results. Therefore, we concluded that to get good results for sticker matching, you must use stickers that show more contrast from the background, unlike yellow on a white background.

6.2. Future work

If we use the circular sampling, radial sampling, and template matching filter from paper [4], we can implement the second approach of using template matching even faster while making templates invariant to scale, orientation, and intensity. In our research, to keep the process unaffected from light, we used matching based on the images' edges, but with the methods in paper [4], voting board sticker detection can be applied without this preprocessing procedure.

References

- [1] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures.," *Mag. Commun. ACM*, vol.15, no. 1, pp. 11-15, 1972.
- [2] T. J. Atherton and D. J. Kerbyson, "Size invariant circle detection," vol. 17, no. February 1997, pp. 795-803, 1999.
- [3] S.E. de Vegt, "A Fast and Robust Algorithm for the Detection of Circular Pieces in a Cyber Physical System", Technische Universiteit Eindhoven University of Technology, 2015
- [4] H. Y. Kim and S. A. de Araujo, "Grayscale, Template-Matching Invariant to Rotation, Scale, Transition, Brightness and Contrast", *PSIVT 2007*, pp. 100-113, 2007
- [5] J. P. Lewis, "Fast Template Matching", *Vision Interface 95*, Canadian Image Processing and Pattern Recognition Society, Quebec City, Canada, May 15-19, 1995, pp 120-123.