Sticker Detection and Counting for Voting Boards

Giyeon Kim, Ilhyun Jo, Seongwon Bang
Seoul National University
Seoul 151-742, Republic of Korea
gompu123@snu.ac.kr
onemonster@snu.ac.kr
makesource@snu.ac.kr

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.

For 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.

Figure 1. Street survey using voting board and stickers

2. Method

2.1. Overview

For this process to work we need to first recognize the different sections of the board. We need to find the area on the board that represents each voting option and count the stickers accordingly. Then we have to know what stickers of each shape and color on the board look like and figure out which stickers to count. There can be some special cases where stickers overlap or stick out of the voting region and we need to add a mechanism to make sure we count overlapping stickers or don't count stickers outside of the voting region by a certain threshold.

2.2. Region Partitioning

The regions of a voting board are usually rectangles. So we are planning to first find and save the data about each region by getting the data of the two corners on the top left and bottom right for each rectangle. We will use that data to infer the region each sticker is placed in and count accordingly.

This method presumes that the picture of the voting board was taken so that the regions came out as perfect rectangles. However the picture might be slanted or have rounded corners. If the picture is slanted, the photo will need to be transformed to be straightened. In the case of rounded corners, the rectangle including the rounded corner rectangle will be used.

2.3. Sticker Detection

Stickers come in many different shapes and sizes. After some brainstorming and research online, we found a few different approaches to solve this problem. The method we end up using will be ultimately decided by testing how well each approach works. The methods range from giving the algorithm a set of default sticker shapes to work with, to using shape recognition technology.

Figure 2. Stickers used for voting boards

The first method we thought of was to precompose a set of stickers and give that data to the algorithm. We thought this was a viable option since the types of stickers people tend to use when creating voting boards were pretty limited. Most people would use circles, and in some very rare cases people would use squares or even worse, stars. So supplying a dataset with these base shapes could be a very viable option. [1] shows how the shape matching could be done.

Another more general approach would be to use circle/polygon detectors. Circle detection can be performed by using Hough transform. [2] shows a way to detect convex polygons. There are a few obstacles with this method such as finding overlapping stickers and finding concave shapes such as stars.

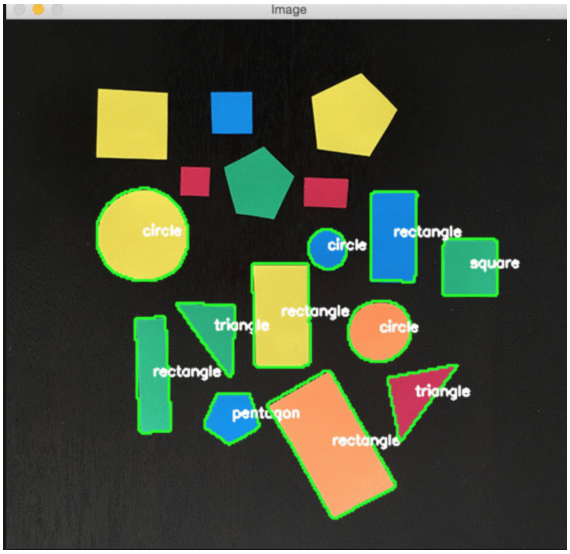

Figure 3. Shape detection for convex polygons

2.4. Error Handling

If overlapping stickers arent countable, our approach for automating sticker counting would be unusable. So we need to create a solution to handle this problem. If we detect stickers using the methods described above, we can detect overlapping by setting a threshold on how similar a shape must be to be recognized as a sticker.

Error handling for stickers misplaced have several different cases to be considered. First of all stickers placed outside of all possible regions should be ignored. If a sticker is on the edges of a regional rectangle, the situation gets a bit tricky. We think that the best way to handle those cases is to put a threshold and include the count only if a sufficient area of the sticker is included.

References

[1] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts, 2002. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[2] Nick Barnes, Gareth Loy, David Shaw, and Antonio Robles-Kelly. Regular polygon detection, 2005. Tenth IEEE International Conference on.