# Mobile Device Camera Connector (Tabiscope)

Paper Seminar by Kyle Wong | Team 7

Partners: Daniel Ahn, Deepak Lingam
Mentors: Dr. Amit Kochhar, Kevin Olds

## 1. Project statement

Our project seeks to create a physical adapter as well as an Android application for connecting an android Tablet to an endoscope in order to allow clinicians to visualize endoscopy in real-time and allow them to take pictures. Currently, imaging in endoscopy is limited by the bulkiness of the endoscopic imaging tower as well as the immense cost of the equipment itself. Our goal is to create a low-cost solution using an Android tablet and an adapter for visualization and imaging during endoscopy that will be usable in third world countries as well as in first-world countries.

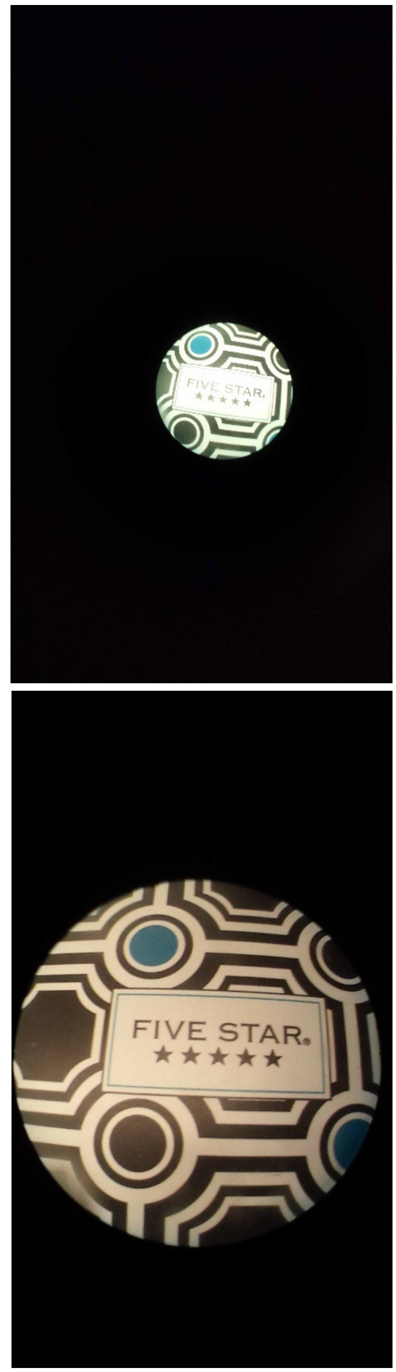


**Figure 1. Example Circle Images**. Taken with Android phone looking through a laparoscope.

## 2. Paper selection

Cuevas, E., Wario, F., Zaldivar, D., & Pérez-Cisneros, M. (2013). Circle detection on images using learning automata. In *Artificial Intelligence, Evolutionary Computing and Metaheuristics* (pp. 545-570). Springer Berlin Heidelberg.

This paper was selected because of its study of how to detect circles in images quickly and accurately which is highly relevant for imaging in endoscopy. The view from an endoscope itself is limited to a circle, and our project requires being able to display that circular image in focus and at a high resolution in real-time. As shown in Figure 1 on the right, being able to identify the circular image and then zooming the camera image appropriately is essential to having usable imaging. Therefore, this paper describes for us a good way to optimize our Android application to identify the endoscopic image.

## 3. Summary of problem and results

### *3.1 The problem*

Detecting circular features is very important for image analysis and for shape recognition. In particular, it has relevance to industrial applications such as automatic inspection of manufactured products and components, aided vectorization of drawings, and target detection. The most common approach is to use variants of the Hough transform on the output of an edge detector, but this requires large storage space and is computationally expensive. This limits the application of most modern circle detectors.

*3.2 The objective*

The authors sought to apply Learning Automata to solve the problem of circle detection and to compare its accuracy and runtime with conventional robust circle detection algorithms including Iterative Randomised Hough Transform (IRHT) and Genetic Algorithms (GA). They aimed to compare these algorithms over a set of natural images as well as a set of synthetic images with noise in order to provide a statistically significant framework for comparing the accuracy and speed of circle detection algorithms. This is shown in Figure 2.

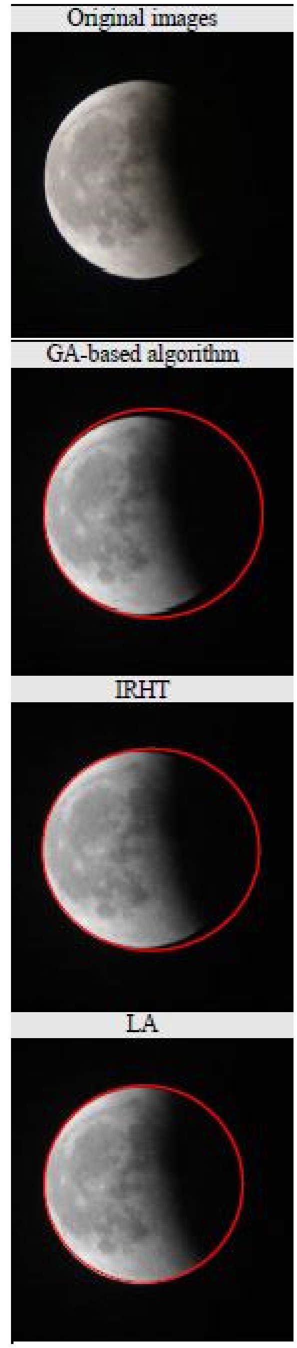


**Figure 2. Natural Image Comparison.**
Cuevas '13

*3.3 The key results*

The authors effectively used Learning Automata (LA) to create a circle detector. Their algorithm ran in a similar fashion to IRHT and GA where images were pre-processed by the Canny algorithm to get a single-pixel edge-only image. The LA algorithm was able to identify circles more accurately and faster than IRHT and GA. Furthermore, the authors showed that their LA algorithm could also work for occluded circles, arcs, and imperfect circles.

## 4. Significance

Rapid and accurate circle detection is an important basic building block for image analysis and shape recognition. Being able to use Learning Automata to quickly and accurately identify circles thus sets a foundation for many other possibilities. Normally, image detection in circles is limited by its large computational cost and slowness. By the speedup and low memory footprint of the LA algorithm, applications that involve tasks like eye-detection or shape recognition can be more widely used. In particular, with the rise of mobile devices with limited speed and memory, many more applications that require circle detection are now more feasible.

## 5. Background

*5.1. Theory of Learning Automata*

Learning Automata solve a form of optimization where they operate in a random environment and uses a decision-making method to progressively improve their performance through a learning process. A probability density function is defined in order to represent actions applied to a random environment. These actions are paired with environment responses, known as the reinforcement signal, which the automata use to update the probability density function in order to select its next action. They thus iterate until an optimal action is found (based on a threshold, or until a certain number of iterations has been reached).

*5.2 Application of Learning Automata to Circle Detection*

In order to detect circles, images were pre-processed by the Canny algorithm to yield a single-pixel edge-only image. Then, only a representative percentage of edge points (about 5%) are considered. Furthermore, the actions for the Learning Automata were represented by all feasible combinations of 3 of these edge points. These 3 points are then used to calculate a potential radius and center of a proposed circle connecting the points as shown in Figure 3.

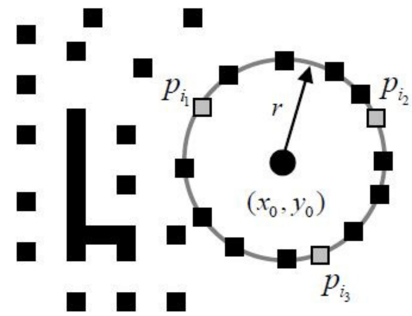


**Figure 3. Circle Proposing**.
Cuevas '13

In order to model the environment's response to an action, a set of points along the proposed circle were generated by the midpoint circle algorithm (MCA) based on the radius and center. Then, the edge-only image is checked to see if these set of points are indeed in the image (coincident), thus verifying that the proposed circle is present in the image.

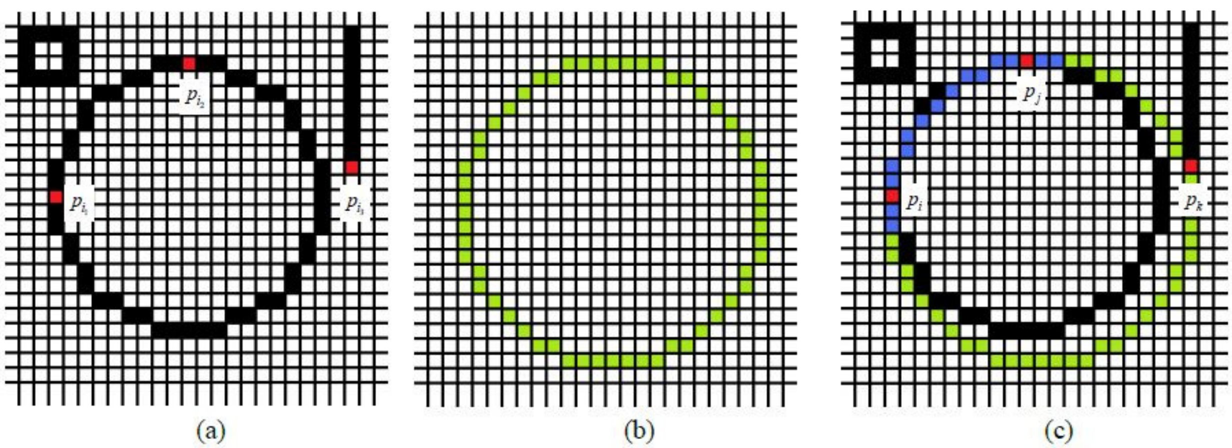


**Figure 4.** Environment reaction to an action $C_i$: The image shown by (a) presents the original edge image while (b) portraits the virtual shape $S_i$ corresponding to $C_i$. The image in (c) shows coincidences between both images through blue or red pixels while the virtual shape is also depicted in green. Cuevas '13

## 6. Materials and Method

The authors implemented the LA algorithm in the programming language C and pre-processed the images with the standard Canny edge-detector from the image-processing toolbox for MATLAB R2008a.

They created 20 synthetic images of 200 x 200 pixels each with one imperfect circle (ellipse shaped) randomly located. Some of the images were generated with noise to increase the complexity of circle detection. They also obtained twenty-five images of 640 x 480 pixels captured with a digital camera under 8-bit colour format with a natural real-life scene that included a circle shape among other objects. They then ran these images through the Canny algorithm to get edge maps.

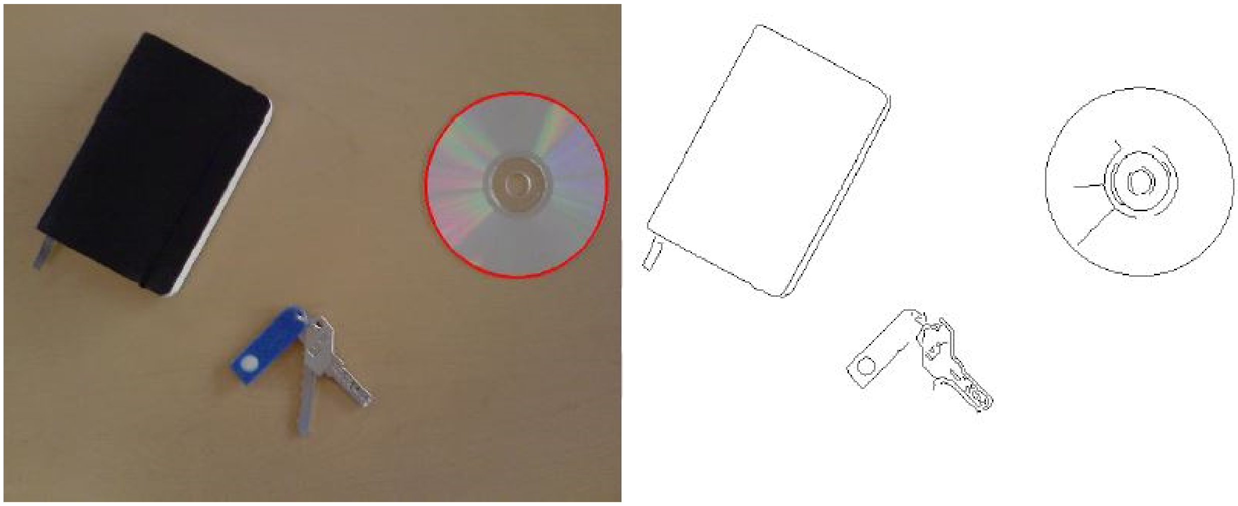

**Figure 5. Natural Image with a variety of shapes and the corresponding edge map.** Cuevas '13

They further generated five synthetic images of 540 x 300 pixels with noise, occluded circles and arcs, varieties of shapes, and multiple circles. They ran their algorithm on these images to explore the capabilities the LA algorithm for circle detection.

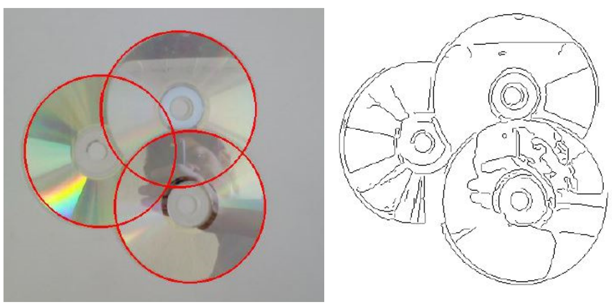

**Figure 6. Natural Image with a multiple occluded circles where the identified circles are labeled in red and the corresponding edge map.** Cuevas '13

The authors then compared the LA algorithm with the IRHT and GA algorithms in terms of accuracy and runtime for all these images.

## 7. Results

| Image | Averaged execution time $\pm$ standard deviation, s | | | Averaged $E_s$ $\pm$ standard deviation | | |
|---|---|---|---|---|---|---|
| | GA | IRHT | LA | GA | IRHT | LA |
| *Synthetic images* | | | | | | |
| (a) | 2.23 $\pm$ (0.41) | 1.71 $\pm$ (0.51) | **0.21 $\pm$ (0.22)** | 0.41 $\pm$ (0.044) | 0.33 $\pm$ (0.052) | **0.22 $\pm$ (0.033)** |
| (b) | 3.15 $\pm$ (0.39) | 2.80 $\pm$ (0.65) | **0.36 $\pm$ (0.24)** | 0.51 $\pm$ (0.038) | 0.37 $\pm$ (0.032) | **0.26 $\pm$ (0.041)** |
| (c) | 3.02 $\pm$ (0.63) | 4.11 $\pm$ (0.71) | **0.64 $\pm$ (0.19)** | 0.71 $\pm$ (0.036) | 0.77 $\pm$ (0.044) | **0.42 $\pm$ (0.011)** |
| *Natural images* | | | | | | |
| (a) | 2.02 $\pm$ (0.32) | 3.11 $\pm$ (0.41) | **0.31 $\pm$ (0.12)** | 0.45 $\pm$ (0.051) | 0.41 $\pm$ (0.029) | **0.25 $\pm$ (0.037)** |
| (b) | 2.11 $\pm$ (0.31) | 3.04 $\pm$ (0.29) | **0.57 $\pm$ (0.13)** | 0.87 $\pm$ (0.071) | 0.71 $\pm$ (0.051) | **0.54 $\pm$ (0.071)** |
| (c) | 2.50 $\pm$ (0.39) | 2.80 $\pm$ (0.17) | **0.51 $\pm$ (0.11)** | 0.67 $\pm$ (0.081) | 0.61 $\pm$ (0.048) | **0.31 $\pm$ (0.015)** |

**Figure 7. Runtime and Accuracy comparison table for GA, IRHT, and the proposed LA algorithm.** Cuevas '13

The authors found that the LA algorithm was both faster and more accurate than the IRHT and GA algorithms at circle detection. They also found that the LA algorithm was robust to noise, could identify circles that were partially occluded, could ignore irrelevant shapes, and could identify multiple circles in a picture.

## 8. Assessment

### *8.1. Positive aspects of this work*

The authors did a very thorough job in the paper of describing their algorithm as well as their methods and how they obtained their datasets. They also created nice summarizing images and a data table to highlight the results of their LA algorithm.

### *8.2. Limitations of this work*

$$E_s = \eta(|x_{true} - x_D| + |y_{true} - y_D|) + \mu|r_{true} - r_D|$$
$$E_s = \text{Error score}$$
$$\eta = \text{weight for accuracy of the center (chosen 0.05)}$$
$$\mu = \text{weight for accuracy of the radius (chosen 0.1)}$$

One issue with the paper is their use of their own constructed measure of accuracy of circle detection relying on the radius and center point listed above which fits perfectly for their algorithm along with the arbitrary weights chosen. Still, the example comparison images they show illustrate the quality of their Learning Automata algorithm.

## 9. Future work and relevance to our project

### *9.1 Future work for the authors*

The authors could compare their LA algorithm to more popular fast, real-time algorithms such as RCD (Randomized Circle Detection algorithm) or the basic Circular Hough Transform that is standardly used. They could have also tested the LA algorithm on a real-time streaming image to prove the speed of their algorithm and to highlight the potential applications of such a fast and accurate circle detection algorithm.

### *9.2. Influence on our project*

For our project, we need to be able to quickly identify the circle of the endoscope image, so we may be able to implement a simple LA algorithm. This will help us in ensuring that we display the image at the right resolution as well as focus the camera on the correct circle of interest automatically in addition to helping us choose the right pixels for adjusting the brightness automatically.

## 10. Conclusions

Research on circle detection and other shape recognition is still an ongoing research topic. However, the proposed LA algorithm provides a nice solution to the circle detection problem. Since so many tasks relate to circle detection, the speed, accuracy, and robustness of this LA algorithm in detecting circles can lead to the improvement of several applications.