Mobile Device Camera Connector (Tabiscope)
Circle Detection with Learning Automata (LA)

Paper Seminar by Kyle Wong

600.446 Computer Integrated Surgery II
Project 7

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

- Design a low cost endoscopic adapter
- Create a system for Android devices
Design Challenge

- Real-time image processing method for Circle Detection for auto-zoom, auto-focus, auto-brightness etc.

goal: automatic

Project 7: Tabiscope (Daniel Ahn, Deepak Lingam, Kyle Wong)
Summary of Problem and Results

- Circle detection quickly and accurately
- Create a set of synthetic and natural images for comparison
- Compare Learning Automata (LA) with Iterative Randomised Hough Transform (IRHT) and Genetic Algorithms (GA)

Cuevas ’13. Circle detection on images using learning automata.

Project 7: Tabiscope (Daniel Ahn, Deepak Lingam, Kyle Wong)
Significance

- LA algorithm works for occluded circles and multiple circles and is robust to noise and fast widespread use.

Cuevas '13. Circle detection on images using learning automata.
Learning Automata (LA) optimize through a learning process.

- Model actions applied to an environment with a probability density function (pdf).
- Pair actions with reinforcement signal to update the pdf to select the next action.
- Iterate until optimal action is found (threshold reached, or number of iterations is done).
Background - Preprocess

- pre-process with Canny Algorithm to get single-pixel edge map; take only a fraction (about 5%) randomly

Matlab - Canny Method
Random Sample
Matlab - Sobel Method
Background - Sample

- sample combinations of 3 points and check

Cuevas ’13. Circle detection on images using learning automata.
Background 3 Circle-find

- Circle calculation

\[
\begin{align*}
    x_0 &= \frac{\text{det}(A)}{4((x_{i_2} - x_{i_1})(y_{i_3} - y_{i_1}) - (x_{i_3} - x_{i_1})(y_{i_2} - y_{i_1}))} \\
    y_0 &= \frac{\text{det}(B)}{4((x_{i_2} - x_{i_1})(y_{i_3} - y_{i_1}) - (x_{i_3} - x_{i_1})(y_{i_2} - y_{i_1}))}
\end{align*}
\]

\[
A = \begin{bmatrix}
    x_{i_2}^2 + y_{i_2}^2 - (x_{i_1}^2 + y_{i_1}^2) & 2(y_{i_1} - y_{i_1}) \\
    x_{i_3}^2 + y_{i_3}^2 - (x_{i_1}^2 + y_{i_1}^2) & 2(y_{i_3} - y_{i_1})
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
    2(x_{i_2} - x_{i_1}) & x_{i_2}^2 + y_{i_2}^2 - (x_{i_1}^2 + y_{i_1}^2) \\
    2(x_{i_3} - x_{i_1}) & x_{i_3}^2 + y_{i_3}^2 - (x_{i_1}^2 + y_{i_1}^2)
\end{bmatrix}
\]

\[
r = \sqrt{(x_0 - x_d)^2 + (y_0 - y_d)^2}
\]
Methods

- Synthetic images with noise
- Real life images

Cuevas '13. Circle detection on images using learning automata.

Project 7: Tabiscope (Daniel Ahn, Deepak Lingam, Kyle Wong)
Methods - LA

- Synthetic images with noise
- Real life images


Project 7: Tabiscope (Daniel Ahn, Deepak Lingam, Kyle Wong)
## Results

<table>
<thead>
<tr>
<th>Image</th>
<th>Averaged execution time ± standard deviation, s</th>
<th>Averaged $E_s$ ± standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GA</td>
<td>IRHT</td>
</tr>
<tr>
<td>Synthetic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>images</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a)</td>
<td>2.23 ± (0.41)</td>
<td>1.71 ± (0.51)</td>
</tr>
<tr>
<td>(b)</td>
<td>3.15 ± (0.39)</td>
<td>2.80 ± (0.65)</td>
</tr>
<tr>
<td>(c)</td>
<td>3.02 ± (0.63)</td>
<td>4.11 ± (0.71)</td>
</tr>
<tr>
<td>Natural images</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a)</td>
<td>2.02 ± (0.32)</td>
<td>3.11 ± (0.41)</td>
</tr>
<tr>
<td>(b)</td>
<td>2.11 ± (0.31)</td>
<td>3.04 ± (0.29)</td>
</tr>
<tr>
<td>(c)</td>
<td>2.50 ± (0.39)</td>
<td>2.80 ± (0.17)</td>
</tr>
</tbody>
</table>

Cuevas '13. *Circle detection on images using learning automata.*
### Results - LA Best


<table>
<thead>
<tr>
<th>Image</th>
<th>Averaged execution time ± standard deviation, s</th>
<th>Averaged $E_s$ ± standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GA</td>
<td>IRHT</td>
</tr>
<tr>
<td><strong>Synthetic images</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a)</td>
<td>2.23 ± (0.41)</td>
<td>1.71 ± (0.51)</td>
</tr>
<tr>
<td>(b)</td>
<td>3.15 ± (0.39)</td>
<td>2.80 ± (0.65)</td>
</tr>
<tr>
<td>(c)</td>
<td>3.02 ± (0.63)</td>
<td>4.11 ± (0.71)</td>
</tr>
<tr>
<td><strong>Natural images</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a)</td>
<td>2.02 ± (0.32)</td>
<td>3.11 ± (0.41)</td>
</tr>
<tr>
<td>(b)</td>
<td>2.11 ± (0.31)</td>
<td>3.04 ± (0.29)</td>
</tr>
<tr>
<td>(c)</td>
<td>2.50 ± (0.39)</td>
<td>2.80 ± (0.17)</td>
</tr>
</tbody>
</table>
Assessment

Positive
- Thorough explanation and dataset results

Negative
- Self-constructed accuracy (arbitrary and biased?)

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

- Compare to other methods like supposedly fast Randomized Circle Detection (accuracy and speed tradeoff)
- Compare to basic Circular Hough Transform (for a baseline)
- Implement in real-time to show capabilities (using real-time Canny edge detector)
Relevance to our Project

- Rapid and accurate circle detection in Endoscopic image

Project 7: Tabiscope (Daniel Ahn, Deepak Lingam, Kyle Wong)
Conclusions

● Circle Detection and shape recognition are still being researched
● Speed, accuracy, memory use, and robustness are vital considerations
● Having a standard of measuring these is essential for benchmarking
  ○ Learning Automata for Circle Detection may be of use for our Endoscope application
Circle Detection with Learning Automata (LA)

- Questions and Feedback?


Project 7: Tabiscope (Daniel Ahn, Deepak Lingam, Kyle Wong)