Vision Based Intelligent Speed Adaptation Through the Use of Mobile Phones

Eilís Rafter(B00020133) Supervisor: Simon McLoughlin

GENERAL PROJECT

Research Question:
Is it possible to develop a vision based solution to accurately predict the speed limit of a stretch of road in a suitable time frame, using a mobile phone?

Technologies:
Software: J2ME, Java, Android API's.
Current technology implemented for the prototype is J2ME with the intention of expanding to Java for use in Android phones.
Hardware: Nokia 5800 *(Current)*, HTC Desire *(Expansion)*.

Deliverables:
Mobile application which will be easily installed on any mobile phone.

Technical Challenges:
- Developing a way to correctly locate red pixels within the data image.
- Finding an efficient algorithm to locate the circle within the data image.
- Developing a way to perform accurate optical character recognition.
- Identifying bottlenecks within the system using the JME SDK 3.0 profiler, determining what is causing them and finding the best optimization algorithms to solve them.

ANALYSIS & DESIGN

Data Acquisition
Acquires a data image through the use of a mobile phone camera.

Red Pixel Detection
Converts Pixels with a Hue > 320°(Red) to white and all other pixels to black.

Circle Detection
Uses a Hough Transform to locate the circle on the binary image.

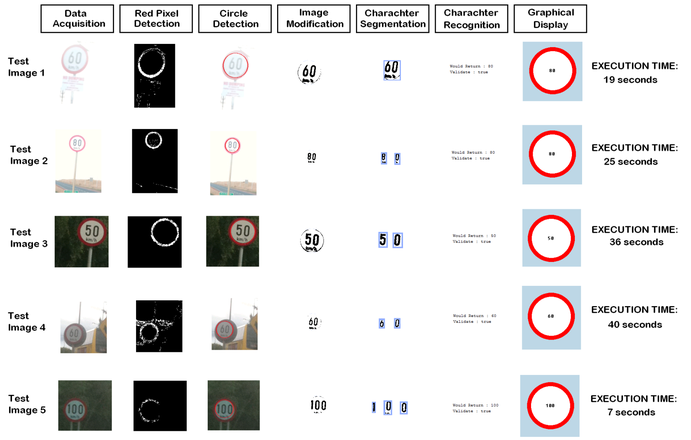
Image Modification
Finds the darkest RGB values and sets pixels near the threshold to black and all other pixels to white.

Charachter Segmentation
Identifies lines of whitespace in the modified image and uses this to separate characters.

Charachter Recognition
Compares segmented characters against a training image and returns string values for identified numbers

Graphical Display
Displays the valid identified string to the user in a speed sign type format.

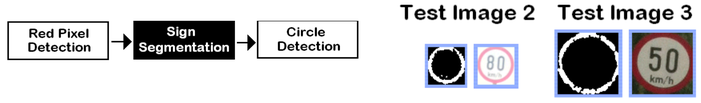
PROTOTYPE



Accuracy	Red Pixels	Circle	Speed Limit	Runtime
Pass	30/39	22/39	21/39	8 seconds
Fail	9/39	17/39	18/39	49 seconds
Average	77%	56%	54%	22 seconds

OPTIMIZATION 1 (Sign Segmentation)

Sign Segmentation was implemented into the project to speed up the process of circle detection. It was inserted into the prototype between red pixel detection and circle detection.



How it Works?
It is able to identify whitespace within the data image and so can separate the road sign easily from the remainder of the image.
This is necessary to pinpoint the near exact radius, which is roughly the width of the new segmented image, whilst also removing the chance of skew from other red objects which may be within the data image.

Benefits:
The benefit to the project of implementing this new method is an overall improvement in execution time. (See Results Table below)

Runtimes	Prototype	Sign Segmentation
Test Image 1	19 seconds	5 seconds
Test Image 2	25 seconds	5 seconds
Test Image 3	36 seconds	9 seconds
Test Image 4	40 seconds	11 seconds
Test Image 5	6 seconds	7 seconds

Risk:
Images must now be of good quality lighting as chance of being accurately segmented is reduced.
This problem can be resolved through the use of Binary Morphology at some point, to join multiple segmented images together to reform the circle, if correct sign detection fails at the first attempt.

DIAGNOSTIC PROFILING

The profiling of the project will be carried out using the JME SDK 3.0 Profiler, the purpose of which is to identify the methods which are decreasing overall performance of the system.

Original Prototype Profile

The profile of the original prototype showed that the main bottle neck was caused by the Circle detection process which needed to be invoked 9 times. This was the main factor in the application being modified to form Optimization 1 (Sign Segmentation).

Hot Spots - Method	Self time [%]	Self time	Invocations
circleDetection.CircleHough.**process** ()		8790 ms (70.9%)	9
ocr.PixelImage.**filter** (int[], int, int)		612 ms (4.9%)	5
com.sun.mmedia.DirectPlayer.**nGetWidth** (int)		450 ms (3.6%)	1
ocr.CreateOCRTrainingImage.**createOCRtrainingImage** ()		410 ms (3.3%)	1
application.Midlet.**drawCircleImage** (int[], int[])		371 ms (3%)	10

Current Implementation Profile

The profile of the current specification shows that the use of Sign Segmentation reduced the time spent on Circle Detection with the need for viewer invocations.

Hot Spots - Method	Self time [%]	Self time	Invocations
com.sun.mmedia.DirectPlayer.**nGetWidth** (int)		30895 ms (68.1%)	1
circleDetection.CircleHough.**process** ()		4507 ms (9.9%)	5
application.Midlet.**createSegments** (int[], java.util.Vector)		2059 ms (4.5%)	4
ocr.PixelImage.**filter** (int[], int, int)		1396 ms (3.1%)	10

Hot Spots - Method	Self time [%]	Self time	Invocations
com.sun.mmedia.DirectPlayer.**nGetWidth** (int)		28456 ms (84.1%)	1
circleDetection.CircleHough.**process** ()		1519 ms (4.5%)	4
ocr.PixelImage.**filter** (int[], int, int)		804 ms (2.4%)	10
application.Midlet.**createSegments** (int[], java.util.Vector)		740 ms (2.2%)	3

Next Implementation Phase

The diagnostic profiling at present indicates that the next stage going forward with implementation will be to attempt to reduce the Circle Detection process to 2 or preferable 1 invocation. Whilst also attempting to reduce time spent acquiring the data image.

PHONE IMPLEMENTATION

Runtime	Red Pixel	Circle	OCR
Single			
Full			

Specs	PC	Nokia 5800	HTC Desire
CPU	2.66 Ghz	11.4 Ghz	1 Ghz
RAM	766 MB	128 MB	576 MB
Runtime	22 sec	S	---

PROJECT PLAN

STAGE 1 Analysis & Design	STAGE 2 Prototype
STAGE 3 Incremental Implementation	STAGE 4 Testing

Current Stage

Stage 4 (Testing)

Tests that will need to be carried out on the final product will be;
- Sequence – Is the application capable of accurately determining the speed limits when differentiating speed signs appear one after the other.
- Road Conditions – Can the application accurately find the speed limit on varying speed sign sizes where the camera may be too close or too far from the actual speed sign.
- Weather Conditions – Can the application determine the speed limit in rain conditions when view is obscured by raindrops and window wipers.
- Lighting Conditions – Is the application capable of finding the speed limit in varying light conditions (Day, Night, Dawn, Dusk)

In addition to basic testing of the application on the Nokia 5800, the same tests will be carried out on the HTC Desire to determine if the overall performance was increased.