Mid-Semester Thesis Report: Submitted in partial fulfillment of the requirements of BITS 422T Thesis

by

© *Nehil Jain* (ID - 2008B4A3560G)

Under Supervision of *Prof. Jeremy R. Cooperstock*

Department of *Electronics and Electrical Engineering*

BITS, PILANI - K.K.BIRLA CAMPUS

May 2013

Nehil Jain McGill University

Abstract

In this report I focus on two aspects of my research projects. First will be the techniques used to enhance Walking Straight Application using Computer Vision. Later, I shall describe the work on Real-Time Emergency Response: Modern Tools for Modern Disasters.

"The purpose of the abstract, which should not exceed 150 words for a Masters' thesis or 350 words for a Doctoral thesis, is to provide sufficient information to allow potential readers to decide on relevance of the thesis. Abstracts listed in Dissertation Abstracts International or Masters' Abstracts International should contain appropriate key words and phrases designed to assist electronic searches."

— MUN School of Graduate Studies

Acknowledgements

Put your acknowledgements here...

"Intellectual and practical assistance, advice, encouragement and sources of monetary support should be acknowledged. It is appropriate to acknowledge the prior publication of any material included in the thesis either in this section or in the introductory chapter of the thesis."

— MUN School of Graduate Studies

Contents

Abstract ii

Acknowledgements iii

List of Tables vi

List of Figures vii

1 Walking straight 1

1.1 Introduction 1

1.2 Walking straight 1

1.3 Are the sensors reliable? 2

1.4 Computer vision solution 3

1.4.1 Problem 3

1.4.2 Colour Segmentation 3

1.4.2.1 Why HSV rather than RGB? 4

1.5 The `Makefile` 5

1.6 Changing Fonts 6

1.7 Accents and Ligatures 6

1.8 Some Lists 6
1.8.1 Subsection 7
1.8.1.1 Subsubsection 7
1.8.1.2 Subsubsection 7

2 Figures and Tables **8**
2.1 Figures 8
2.2 Tables 12

Bibliography **15**

A Appendix title **16**

List of Tables

2.1 Fall Semester Enrollment 13

2.2 Masters Degrees Conferred by Convocation Session — 1950 to 2009 . 14

List of Figures

2.1 This is MUN's logo 8
2.2 MUN Fall Enrollment 2005 – 2009 9
2.3 MUN Fall Enrollment 2005 – 2009 (landscape) 10
2.4 MUN Fall Enrollment 2005 – 2009 (rotated) 11
2.5 A deadlocked Petri net 11
2.6 Hello World 12

Chapter 1

Walking straight

1.1 Introduction

In this report I focus on two aspects of my work. First will be the techniques used to enhance Walking Straight Application using Computer Vision. Later, I shall describe the work on Real-Time Emergency Response: Modern Tools for Modern Disasters.

1.2 Walking straight

Walking without vision, results in veering, an inability to maintain a straight path that has important consequences. This is a well-known problem. Many studies have been done on this and proved the veering effect. The reason is not certain but is hypothesized to be motor error in stepping movement. Blind people have the same problem. This makes crossing intersections and walking straight dangerous and unsafe.

Walking Straight a application for the smartphone device was developed.[3] It

uses the available hardware on a smartphone device, the iPhone 4 to provide real-time audio feedback to the blind users to correct their deviation. The application is built using the ISAS (In-Situ Audio Services) [1] system architecture. The application uses compass and gyroscope sensors to calculate the deviation from a straight path. The values from the sensors are filtered to remove body sway of a person while walking. Different auditory feedback designs were evaluated experimentally and their performance was evaluated. A continuous tone played in the ear in the side of the deviation is concluded to be the most efficient way of rendering the audio feedback to the blind. [3].

1.3 Are the sensors reliable?

The reliability of todays smartphone sensors was done.[4—-sensors study] The current state of walking straight algorithm assumes a stable 10Hz Gyroscope. It solely depends on the sensors to calculate the heading. Captoins is mising The other two sensors are Compass and Gyroscope. Figure 2 indicates compass values (cyan) against ground truth (black), the latter of which is constant (horizontal) for each straight-line leg of the walk. Actual compass error (red) is calculated as the absolute difference between these two, whereas reported compass error (grey) is an estimate of error magnitude by the sensor itself. As can be seen, the actual error fluctuates both above and below the estimate. Gaps in the plot represent transitions between legs of the walk, during which we have no ground truth heading information.

Yaw (green), obtained from the gyro sensor, is not calibrated to north, so this only represents relative variation. This data is should be a flat line, excepting body

sway while walking. Slope in yaw indicates drift, observed in all legs of this walk. The reported course (purple) is derived from the direction of travel based on previous location updates. Both the iPhone and Android devices report course and speed based on location changes, but these appear to be of limited use even in the constrained straight-line testing we performed. [2] The inaccurate results of this study point that our application cannot reply on these sensors solely.

1.4 Computer vision solution

I have been working on overcoming the sensor inaccuracies using the camera in the smartphone and using computer vision techniques to generate a better feedback for the blind than what is present currently.

1.4.1 Problem

The problem that I am solving is, to help blind users safely cross the street at intersections. In this problem, the sub problems are when? to cross it and then get a feedback to cross it within the pedestrian markings by walking straight.

The correct time of crossing the street can be determined by the state of the traffic light. The Traffic light recognition can be done using Color Segmentation, Shape Segmentation or template Matching or a combination of these.

1.4.2 Colour Segmentation

Colour Segmentations is the technique to partition into different sets of pixels. This can be done using specific color of the object we want to identify in the frame. I am

using OpenCV 2.4 to do this. I first calculate the Hue, Saturation, and Value(HSV) Model. The image is from the source is in RGB format. It is converted to HSV and then standard thresholding technique is applied. HSV is color based model on human vision. The image on the right is the Binary image, the red cup being shown with white pixels.

1.4.2.1 Why HSV rather than RGB?

- The simple answer is that unlike RGB, HSV separates Luma, or the image intensity, from Chroma or the colour information. This is very useful in many applications. For example, if you want to do histogram equalization of a colour image, you probably want to do that only on the intensity component, and leave the colour components alone. Otherwise you will get very strange colours.

- In computer vision you often want to separate colour components from intensity for various reasons, such as robustness to lighting changes, or removing shadows.

- If you run into a problem, Google may be a helpful resource.

- Concentrate on content, let LaTeX handle the typesetting.

- Don't worry about warnings related to:

  - overfull `hboxes/boxes`

  - underfull `hboxes/vboxes`

  These can be corrected with modest rewording of your text prior to submission of your final copy.

1.5 The `Makefile`

You can use `make` to "build" your thesis on the Linux command line[1] This will automatically run the `bibtex` program to create your bibliography and will also re-run `latex` as necessary to ensure that all references are resolved. A device independent file (`thesis.dvi`) will be created, by default. If you are using this template in another environment other than the Linux command line, then the `Makefile` will probably not be useful to you.

- To make a PostScript copy of your thesis, type the following at the command line:

  ```
  make thesis.ps
  ```

- To generate a PDF copy of your thesis, run:

  ```
  make thesis.pdf
  ```

- To generate a PDF/A-1b copy of your thesis (which should satisfy the SGS's ethesis submission requirements):

  ```
  make ethesis.pdf
  ```

- To remove all the files generated by `bibtex` and `latex`, use the command:

  ```
  make clean
  ```

- To remove the intermediate files, but leave the PostScript and DVI/PDF files intact, use the command:

  ```
  make neat
  ```

[1] Linux is available on all machines running LabNet in *The Commons* and in other computer labs on campus.

As you add or remove figures, chapters, or appendices to your thesis, make sure you keep the `Makefile` upto date, too (see the `FIGURES` and `FILES` macros in the `Makefile`).

1.6 Changing Fonts

Change fonts: Large, `verbatim ~@#$%^&*(){}[]`, Small Caps, *slanted text*, *emphasized text*, `typewriter text`.

1.7 Accents and Ligatures

Some accents: é è ô ü ç ï í ñ ā ă ă

Some ligatures: flæffi

1.8 Some Lists

Here is a nested enumeration:

1. An enumerated list of items.
   (a) which can
   (b) nest
      i. to arbitrary
      ii. levels
2. More items

3. in the top

4. level list.

Another enumeration:

1. (a) Main 1 part 1

   (b) Main 1 part 2

2. (a) Main 2 part 1

   (b) Main 2 part 2

1.8.1 Subsection

1.8.1.1 Subsubsection

This section is referred to by Section 1.2.

1.8.1.2 Subsubsection

<Empty subsection>

Chapter 2

Figures and Tables

2.1 Figures

We can include encapsulated PostScript™ figures (`.eps`) in the document and refer to it using a label. For example, MUN's logo can be seen in Figure 2.1.

Figure 2.1: This is MUN's logo

Figure 2.2 shows a chart of MUN's Fall enrollment from 2005 – 2009.[1] The figure

[1]From *Memorial University of Newfoundland — Fact Book 2009.*

Figure 2.2: MUN Fall Enrollment 2005 – 2009

was created using the Calc spreadsheet application of the office suite OpenOffice.org.[2] This figure was reduced by 50%.

For larger figures, we can use landscape mode to rotate the page and display the figure using the `\munlepsfig` command, as shown in Figure 2.3. The figure will be the only thing on the page when typeset in landscape mode. (The figure is reduced to 85% of its original size.)

Alternatively, if we just want to rotate the figure, but not the entire page, we can specify an `angle` attribute in the default argument of the `\munepsfig` command. The result is shown in Figure 2.4. If the figure is too large or if there isn't sufficient text, then the figure may appear on its own page.

Note that all three of the enrollment figures are basically the same file, but with different names — on Linux, they are symbolic links to the same file. The filenames

[2]This office suite can be downloaded at no cost from `http://openoffice.org/`. Unlike other commercial office suites, OpenOffice.org may be legally shared with colleagues and fellow students. There are versions for Linux, Microsoft Windows, Mac OS X and Solaris. Also, unlike commercial offerings, OpenOffice.org does not require activation using registration keys.

Figure 2.4: MUN Fall Enrollment 2005 – 2009 (rotated)

have to be different because the reference labels need to be unique.

Figure 2.5 shows a Petri net created using the `xfig` program (`http://www.xfig.org/`) which has very good support for LaTeX. This figure has been reduced to 40% of its original size.

Figure 2.5: A deadlocked Petri net

We can also create figures of text (such as short code snippets) using the `\muntxtfig` command, as show in Figure 2.6.

```
#include <stdio.h>

int main(int argc, char **argv)
{
  printf("Hello world!\n");
  exit(0);
}
```

Figure 2.6: Hello World

2.2 Tables

We can also create tables, as seen by Table 2.1. Note that, as required by SGS guidelines, the caption for a table appears above the table whereas figure captions appear below the figures. Tables and figures can "float" — they may not appear on the page on which they are mentioned. LaTeX tries to handle figure and table placement intelligently, but if if you have a lot of them without a reasonable amount of surrounding textual content, the figures and tables can accumulate towards the end of the chapter. Generally speaking, if there is sufficient text explaining the tables and figures or if the tables/figures are relatively small, this may not be a problem. However, if you have a lot of tables or figures, it may be a good idea to put them in an appendix and refer to them as the need arises.

Table 2.2 shows a different table in landscape mode.[3] This is useful if your table

[3]This data was also taken from the *Memorial University of Newfoundland — Fact Book 2009*.

Table 2.1: Fall Semester Enrollment

	Undergraduate			Graduate		
	F/T	P/T	Total	F/T	P/T	Total
2004	13,191	2,223	15,414	1,308	879	2,187
2005	13,184	2,143	15,327	1,375	920	2,295
2006	12,809	2,224	15,033	1,373	899	2,272
2007	12,634	2,155	14,789	1,403	899	2,302
2008	12,269	2,208	14,477	1,410	1,005	2,415
2009	12,382	2,323	14,705	1,567	1,106	2,673

is too wide for the page. Tables are double-spaced by default. To single-space a table, change the `\baselinestretch` before beginning the table environment. Remember to restore it after the environment has ended.

Table 2.2: Masters Degrees Conferre

	2009		2
	May	Oct	Ma
Degrees			
Master of Applied Science	14	2	1
Master of Applied Social Psychology	1	5	
Master of Applied Statistics	0	0	
Master of Arts	37	49	2
Master of Business Administration	14	16	2
Master of Education	107	87	12
Master of Employment Relations	8	9	
Master of Engineering	20	19	2
Master of Environmental Science	3	3	
Master of Marine Studies	2	0	
Master of Music	4	1	
Master of Nursing	7	8	1
Master of Oil and Gas Studies	0	0	
Master of Philosophy	5	4	
Master of Physical Education	0	2	
Master of Public Health	0	8	

Bibliography

Appendix A

Appendix title

This is Appendix A.

You can have additional appendices too (*e.g.*, `apdxb.tex`, `apdxc.tex`, *etc.*). If you don't need any appendices, delete the appendix related lines from `thesis.tex` and the file names from `Makefile`.