2008-01-01

Evaluation of an Image Processing Algorithm for Scene Change Detection

Daniel Flores
University of Texas at El Paso, floresdx@sbcglobal.net

EVALUATION OF AN IMAGE PROCESSING ALGORITHM FOR SCENE CHANGE DETECTION

DANIEL FLORES

Department of Electrical and Computer Engineering

APPROVED:

_____
John A. Moya, Ph.D., Chair

_____
Eric MacDonald, Ph.D.

_____
Ruey (Kelvin) Cheu, Ph.D.

_____
Patricia D. Witherspoon, Ph.D.
Dean of the Graduate School

EVALUATION OF AN IMAGE PROCESSING ALGORITHM FOR SCENE CHANGE DETECTION

by

DANIEL FLORES, B.S.E.E.

THESIS

Presented to the faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

December 2008

ACKNOWLEDGMENTS

"Despite the great advances made in science and technology, each day we see how much suffering there is in the world on account of different kinds of poverty, both material and spiritual... Science can contribute greatly to making the world and mankind more human... It is not science that redeems man: man is redeemed by love..." Benedict XVI.

Thanks to God for granting me the chance and the ability to successfully complete this study.

I wish to express my deepest gratitude to my advisor Prof. John Moya for his valuable advice and guidance of this work and for giving his valuable time, despite of his tight schedules.

I also want to express my gratitude to the official referees of my thesis defense Dr. Eric MacDonald and Dr. Kelvin Cheu (University of Texas at El Paso) whose comments and criticism were helpful in refining the draft version of the thesis into its final form. I also want to express my thanks to my sister Dr. Araceli Flores for revising the English language and for her valuable comments and criticism during its preparation of the manuscript.

I am also thankful to the numerous individuals who have directly or indirectly contributed to the completion of this work.

Finally, I am particularly and deeply grateful to my wife, Mariela Loya-Flores, for helping and assisting me in all the stages of this work. Without her help and comprehension this study would never have been possible. Her constant and continuous support proves her love during the whole course of this quest. Special thanks to my daughters Fátima and Cristina, and my sons Pablo Daniel and Santiago José for their love, patience, company, inspiration and all kinds of support during all of my studies.

Daniel Flores

El Paso, Texas

November 2008

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES (1/2)

LIST OF FIGURES (2/2)

CHAPTER I

INTRODUCTION

Automobile manufacturers are becoming more interested in offering safety in their vehicles without added significant cost. They are interested in providing safety features which appeal to the consumer (driver/passenger) and that protect the public in general. Also, government transportation agencies at the Federal, State and Local level (e.g. the Texas Department of Transportation, TX DOT) are working to implement and provide information to drivers in order to help them make better decisions before and while driving. Some of the efforts that government transportation agencies are making are the implementation and investment on the use of technology on the road, this field is known as Intelligent Transportation Systems (ITS). One example of ITS is the Dynamic Message Signs (DMS) which are big text boards located on the middle of the highway. They provide information about existing traffic conditions or future roadwork. Another type of ITS is the Remote Traffic Microwave Sensors (RTMS) which are microwave devices located on poles next to the highway. RTMS provide information about vehicles speed, traffic delays and vehicle classification. A third type of ITS is the Highway Advisory Radio Systems (HAR), which are a radio station network broadcasting usually via Amplitude Modulation (AM) and located along the highway. The HAR allows any driver to tune to that station, and provides information about traffic, road conditions, and

future closures due to major construction. A fourth type of ITS is Closed Circuit TV cameras (CCTV) which are video cameras located on strategic points along the highway and main city streets. The CCTV's serve as a video surveillance device that helps a traffic management center and emergency agencies to have a better idea of what is happening in the field in order to make decisions and implement emergency plans on traffic signals. The use of advanced technology has been supported by various researches including Michalopoulos [1] who has emphasized the importance of implementing advanced technology and equipment in the traffic field.

As previously mentioned, some local governments, like the City of El Paso, are also utilizing ITS on city's streets. For example, a microcomputer traffic controller controls signalized intersections, which are then able to store different traffic synchronization plans and make basic timing decisions according to settings previously programmed by an engineer. These types of controllers can use sensors to obtain real-time information from vehicles or pedestrians and are able to make decisions about the distribution of green time.

The most popular sensor is the inductive loop detector type [2]. It is installed underground and is used by a traffic signal controller to detect when a vehicle is waiting for a green light, or to extend the green light, if needed, according to the traffic volume, on other words, the traffic controller responds to the needs of traffic [2]. Video Detection is another type of sensor, which has been

previously addressed by Michalopoulos [1] and Chartziianoun, Hockaday, Kaighn and Ponce [3]. Video traffic detection is a newer type of technology that replaces underground inductive loop detectors. This sensing scheme is versatile and easy to install. It provides video feed which can be sent to the traffic management center to monitor the traffic at an intersection (Figure 1). It is a visual aid, which is a good tool for traffic management personnel in order to operate the intersection and to perform a large variety of traffic studies.

It is evident that electronic technology is being used in the traffic and

Figure 1. Video detector snap shot.

transportation field. Further, as long as the technology keeps on developing and improving, more applications and more effective ways to apply these improvements in this field will be implemented.

PROBLEM STATEMENT

Despite all the efforts that government agencies are making, the number of accidents is still considerable. For example, within the city of El Paso in 2007 alone more than 16,000 accidents were reported to El Paso Police department, almost an accident per every 44 residents of the city unfortunately, this also resulted in an average of almost one fatality per week.

A recent study done by the Institute of Transportations Engineers (ITE) [4] reveals that it is estimated that there were 218,000 red-light running crashes at intersections in the United States during the year 2002. These crashes resulted in 181,000 injuries and 880 fatalities. Seyfried [4] stated that red-light running, along with other aggressive driving behaviors, has become a national highway safety problem, and according to him, 63 percent of Americans witness a red-light running incident more than once a week.

Similarly, the Texas Transportation Institute (TTI) [5] has stated that red-light violations have a societal cost to Texans of about $2.0 billion dollars each year. This includes the direct cost of the crashes (i.e. property damage, medical costs, and legal fees) as well as indirect costs associated with lost earnings and a reduced quality of life. The direct cost to Texan motorists was estimated at $1.4 billion annually.

So far, in this document, only external devices that provide the driver with information or a more efficient traffic flow have been discussed. However, more can be done for efficiency and safety. Wright and Baker [6] pointed out that traffic accidents are caused by human factors, vehicle factors and environment factors that entail road conditions, inadequate infrastructure design, weather conditions and the driver behavior. Some research studies, including the ones reported by Hall [7], have shown that the driver causes most of the traffic accidents. This occurs especially when driving under influence of drugs or alcohol, driving aggressively, speeding or being distracted (i.e. using a cell phone, eating, etc).

Government agencies and automobile manufacturers are also aware of the problem, and they are starting to equip their products (vehicles) with sophisticated and state-of-the-art technology tools to help the driver drive more safely, efficiently and comfortably. For instance, the European automotive industry has established the advanced driver assistance system (ADAS); this technology helps drivers to avoid or mitigate accidents through the use of in-vehicle gadgets that sense the nature and significance of a danger, while also taking the driver's state into account [8]. Another good examples would be the integration of a Global Positioning System (GPS), Satellite Radio (XM), temperature sensors, voice recognition, voice activated functions, and crash support communication centers, among others. The aforementioned are some examples that show the implementation of the technology on a vehicle is not new

field and that it is growing and improving quickly. This is turning vehicles into more sophisticated devices, with the capability of perform a wide variety of functions.

As discussed earlier, traffic accidents are caused by different factors, including roadway conditions, speeding and driving behaviors (driving under the influence of alcohol or drugs, and distractions while driving caused by the use of cell phone, changing radio stations, eating, etc.). Hence, it is important to provide the driver with tools and options to increase safety that will help reduce the risk of getting involved in an accident. For example, the installation of a video camera on a car can help the car to watch for problem by analyzing the video and assisting the driver to make good travel decisions.

Moya and Saenz [9], reported such an alternative to help the driver to keep track of multiple moving objects and people in real scenes via integration of a video device and a vehicle. This can be achieved through the adaptation of an existing approach, inspired by the fly's eye, which consists of a real time scene-change detector that could be used in several traffic situations [10].

Although the primary goal of the work here is to develop techniques to be used on traffic situations, due to the complexity of the traffic application (the amount of changes and moving objects is a sequence of images, and the multiple background changes), it has been decided to start monitoring changes on Printed Circuit Boards (PCBs). By keeping track of the disappearance or

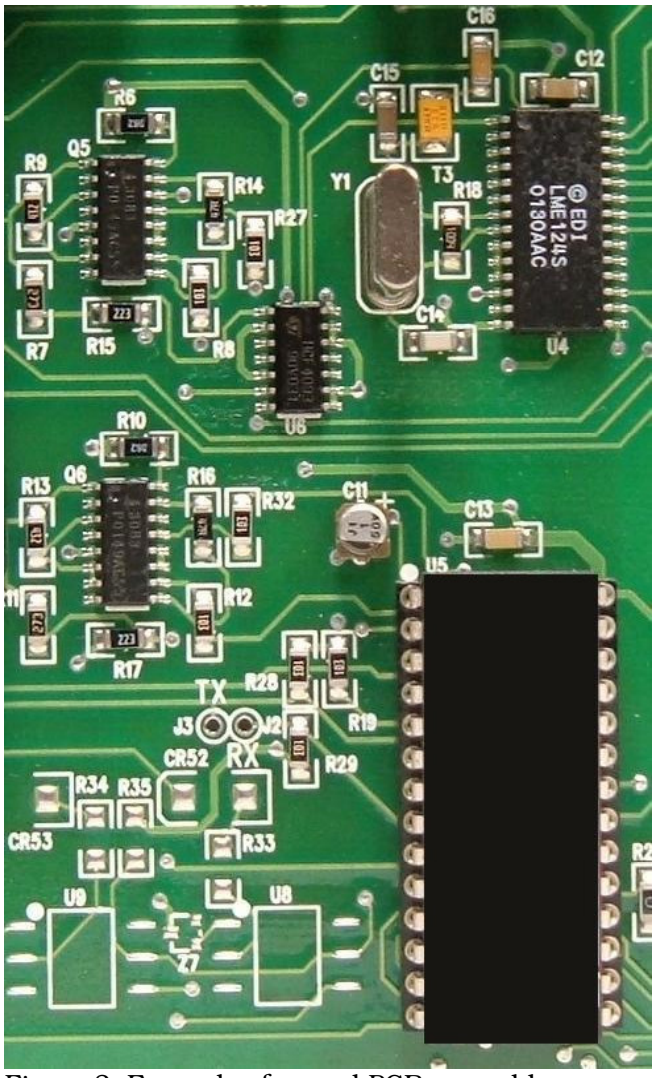

Figure 2. Example of a good PCB assembly.

appearance of objects or changes in position of objects on a PCB, tasks similar to those required during traffic monitoring will be addressed [10]. For instance, the good reference PCB image [figure 2) is equivalent to an original background for a traffic situation; any change from the original image is equivalent to an object that appears on the traffic scene, i.e. moving object on the road.

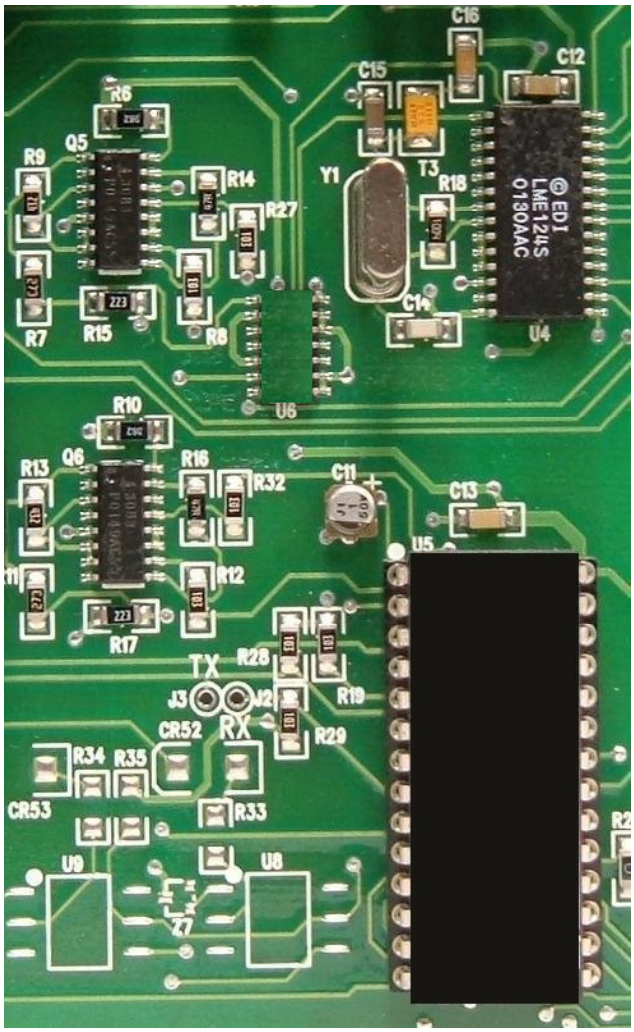

Figure 3. Example of a PCB with an assembly error, U6 is missing.

Several techniques for Printed Circuit Board (PCB) inspections exist in the manufacturing field, some being automated and others being human-based [10]. Regardless of the technique, PCB inspections play a key role for quality control and assure manufacturer's requirements are met via a fast, reliable and accurate approach.

Full human–based PCB inspection techniques turn out to be a demanding and exhausting job which may lead to confusion and to production issues. For instance, a human-based inspection requires an individual to quickly inspect a PCB by comparing it to an original good assembly and noting any errors. This can be inaccurate, slow, and more costly when compared to and automated process [11].

Automated PCB inspection can be very helpful for large-scale manufacturer's assembly lines. The novel technique proposed by Moya and Saenz [10], suggests using a global, just-enough-smart, quickly updated, automated technique with human inspection. This approach consists of a video camera and program that can be easily installed and configured on the assembly line to start inspecting PCBs, pointing to the areas that require special attention by the operator. Given its basis in flies which essentially are visual scene change detectors, the correct implementation of this technique may help to reduce time and money because it is known that up to 45% of printed circuit board faults are caused by component misplacement or absence [figure 3] [11].

The fly's vision system consists of retina, lamina, medulla and lobula it has superior movement detection ability [13]. Further, flies rely heavily on sight for survival. The compound eyes of flies are composed of thousands of individual lenses and are very sensitive to movement. Some flies have very accurate 3D vision.

A compound eye is a visual organ found in arthropods such as insects and crustaceans. It consists of one to thousands of ommatidia which are tiny independent photoreception units that consist of a cornea, lens, and photoreceptor cells which distinguish brightness and color [14, 15]. The image perceived by the arthropod is a combination of inputs from numerous ommatidia which are oriented to point in slightly different directions. Compared with single-aperture eyes, compound eyes have poor image resolution; however, they possess a very large view angle and the ability to detect fast movement and, in some cases, the polarization of light [15].

An ommatidium contains a cluster of photoreceptor cells surrounded by support cells and pigment cells. The outer part of the ommatidium is overlaid with a transparent cornea. Each ommatidium is innervated by one axon and thus provides the brain with one picture element. The brain forms an image from these independent picture elements. The number of ommatidia in the eye depends upon the type of insect and ranges from just a handful in the primitive Archaeognatha and Thysanura to several hundred in larger Diptera.

Each ommatidium is hexagonal in cross section, and is ten times longer than its diameter. The diameter is largest at the surface, tapering toward the inner end. At the outer surface there is a cornea, below which is a pseudocone which acts to further focus the light. The cornea and pseudocone form the outer 10% of the length of the ommatidium.

The visual system of the fly, briefly explained before, inspired the tracking model proposed by Moya and Saenz [9, 10, 12]. This is not an exact model of the fly's eye function but it uses its principles for the tracking model explained later. The neuro-ommatidium, provides the most interesting biological concepts that is interpreted by Saenz [12] with mathematical and computer concepts.

BACKGROUND

The tracking algorithm proposed by Moya and Saenz, similar to the neuro-ommatidium of the fly's eye, it is proposed in [10, 12] that a tracking approach use a set of image powers $\{P_m(t)\}$ collected via four normalized Gaussian-shaped position sensitivity function $g_m(i,j)$, for m=1 to 4 and a total power of the image $f(i,j,t_k)$ given by

$$P_T(t_k) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [f(i,j,t_k)]^2 \tag{1}$$

where N is the size of the N x N image and k = *0,1,2...etc.*, depending on the number of sequential images. The power resulting via a point-wise multiplication of an image $f(i,j,t_k)$ by each $\{g_m(i,j)\}$ is determined by

$$P_m(t_k) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [g_m(i,j) f(i,j,t_k)]^2 \tag{2}$$

where $\{g_m(i,j)\}$ are non-concentric filter (four filters, m = 1 to 4, see figure 7).

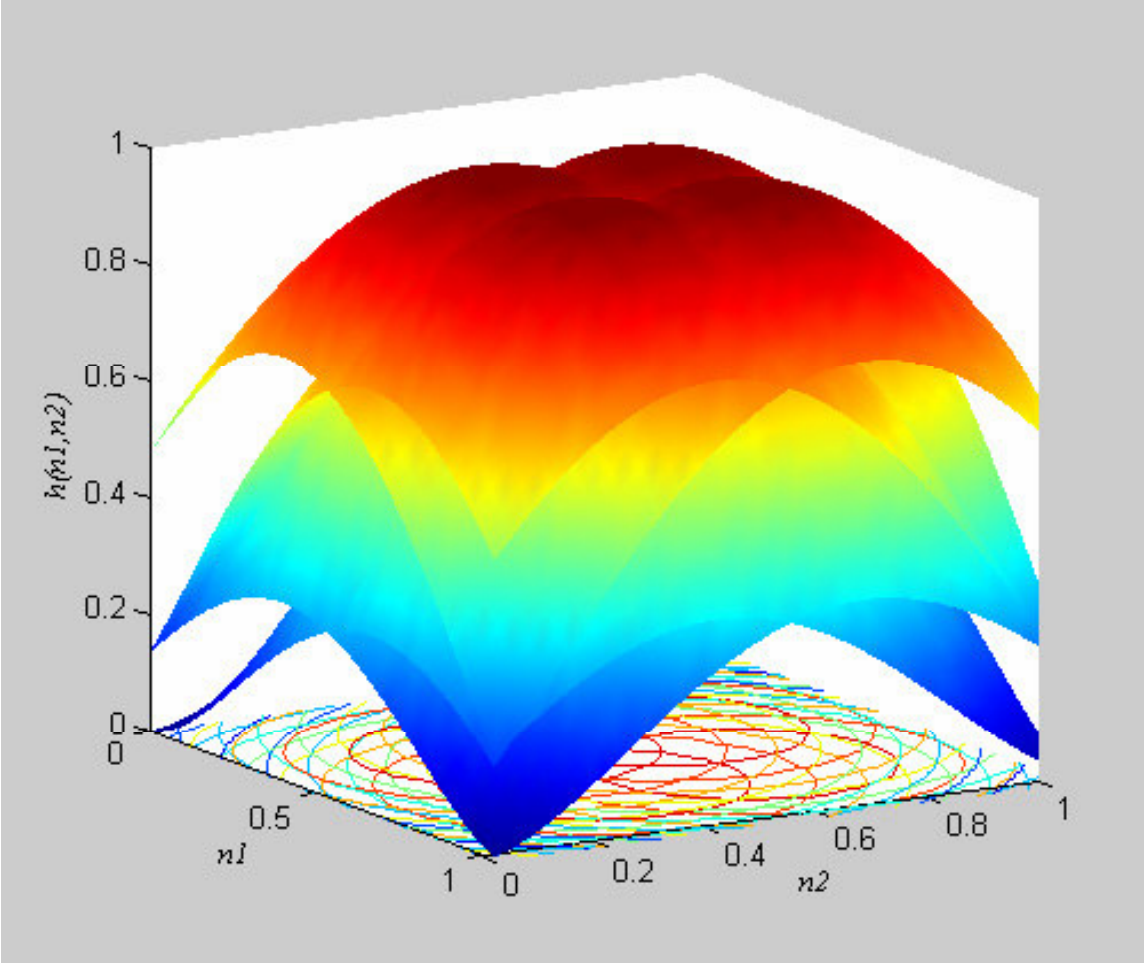

Figure 4. 3D view of Gaussians.

In this case, for mathematical reasons, each function $\{g_m(i,j)\}$ is a shifted version of the Gaussian-approximating function

$$h(i,j) = 1 - sin\left(\left(\frac{i-\frac{N}{2}}{C}\right)^2 + \left(\frac{j-\frac{N}{2}}{C}\right)^2\right) \qquad [3]$$

evaluated from *0* to *N-1* for both *i* and *j* (figure 5).

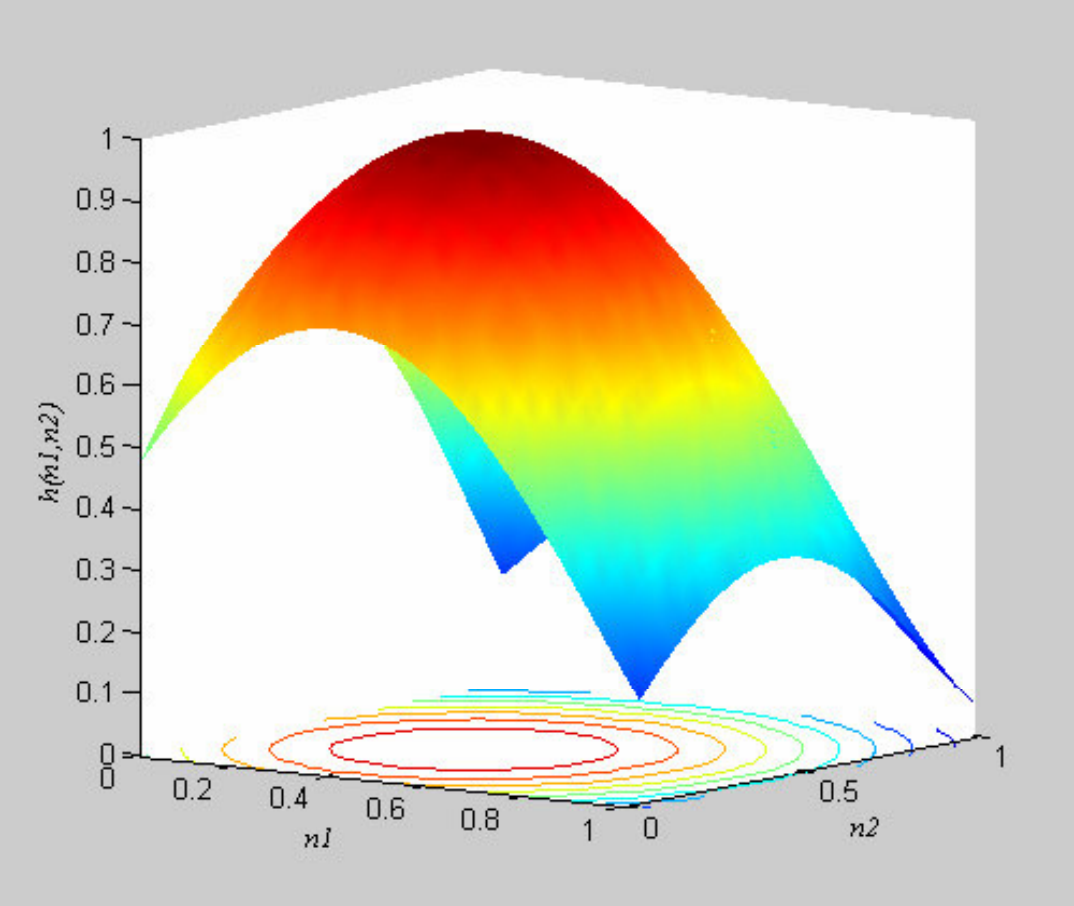

Figure 5. 3D view of one Gaussian.

Upon knowing the power values over time for a sequence of the images, the radial position of a single object (referenced to the background) can be found using

$$R_m(t_n) = \sqrt{2}N(1-u_m)\left[1-\frac{1}{\pi}cos^{-1}\left[1-\frac{2y(t_n)}{A(t_n)}\right]\right] \qquad (4)$$

where N is the length of an image side; u_m is the shortest, normalized diagonal distance from a corner in the image to the center of the Gaussian filter $g_m(i,j)$; $y_m(t_n)$ is the absolute change in the power P_m received via $g_m(i,j)$ and given by

$$y_m(t_n) = \left| P_m(t_n) - P_m(t_0) \right| \; ; \tag{5}$$

and $A(t_n)$ is an estimate of the maximum power change, where

$$A(t_n) = P_u(t_n) \frac{\left| P_u(t_n) - P_u(t_0) \right|}{P_u(t_0)} \tag{6}$$

$A(t_n)$ can be calculated using the power P_u collected from $h_{u(i,j)}$.

Knowing the latter radii, an estimate of the position of the moving object can then be determined using triangulation. These estimated positions can be combined into one position using a data diffusion technique [10, 12].

ALGORITHM

Based on the algorithm proposed by Moya and Saenz [10, 12), and stated in the formulas 1 to 6, a set of Matlab files have been developed.

The algorithm process, as explained briefly in the flow chart on figure 6, starts with the reading of the original image, which is used as a background, and the image that will be analyzed. Once the images are stored in memory, the filters calculations are performed, applying the Gaussian filters. Via inverse powers one may know the radii information for the image change and calculate the position of the change, by performing a triangulation calculation by means of a Law of Cosines calculation. At this point, estimation for the position of the object is already known, and then only placing a mark on the image is needed. This marking procedure is done by placing a red crosshair on the image.

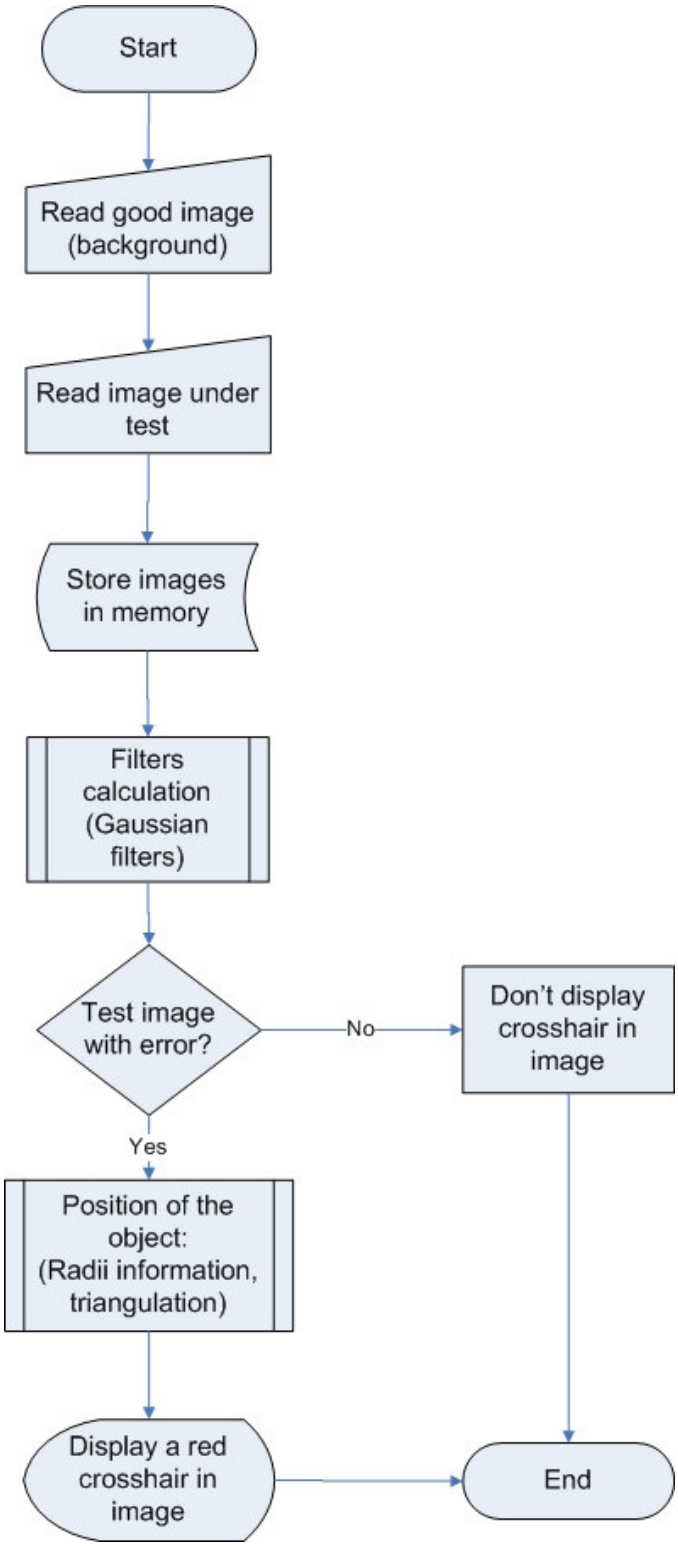

Figure 6. Flow chart showing the algorithm process of the Matlab scripts for the image processing.

THESIS SUMMARY

This thesis analyses the functionality of this new scene-change detector concept that could be used in position sensing of objects on Printed Circuit Boards. Afterwards the same algorithm can be implemented on real-time traffic situations.

Chapter 2 presents a summary of data set obtained from the algorithm validation experiment. Chapter 3 provides concluding comments and possible future efforts.

CHAPTER II

SUMMARY OF DATA SET

In order to evaluate and validate the algorithm studied in this thesis, a group of Matlab scripts were created based on the adapted algorithm discussed before. The methodology applied in the evaluation process is as follows:

- Identify a group of PCBs that are operational and fully assembled and that are a similar model. Create errors on those images, manipulating the images by software. The errors were based on real manufacturing assembly problems, such as: missing and misplaced components.

- Process in all the images with errors via the Matlab scripts and compares their results with the process results from the original images with no errors. The results were also visually checked to see if the software is identifying all the errors created intentionally for this specified purpose.

- Run the Matlab scripts using only vehicular traffic images and see how the algorithm detects a vehicle on the intersection.

Three different groups of PCBs were used for this experiment: Ten loop detector cards; ten modem card and three S12 cards. Using each of the loop

detector and modem images five error images were generated. Similarly, six error images were made using the S12 card. In total, for this first step one hundred and fifteen (115) images with errors were used on this experiment. Figures 8, 10 and12 show some examples of errors per each model of card used on this experiment.

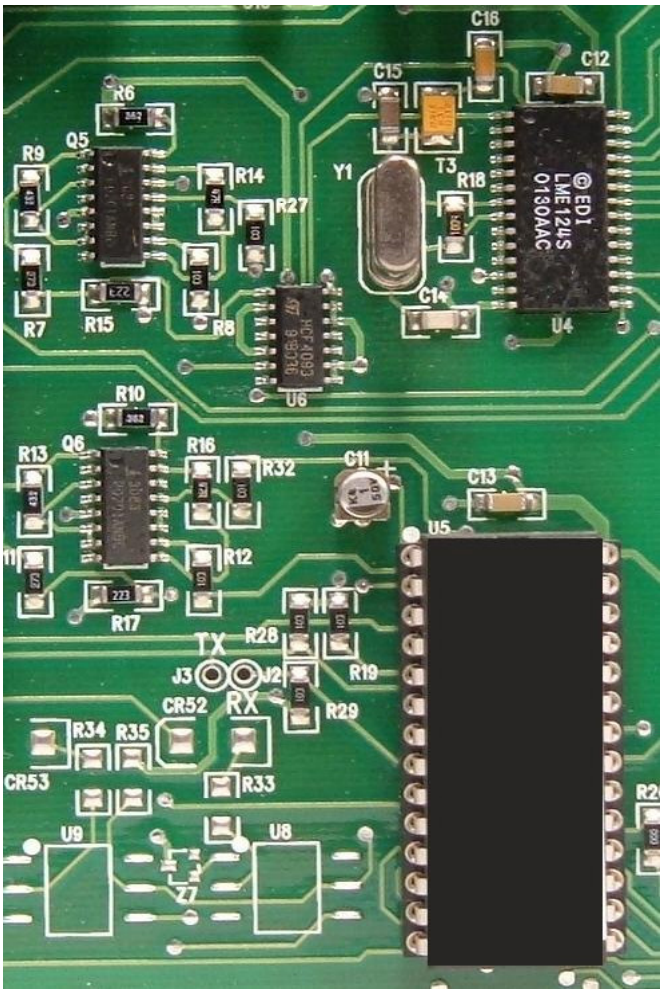

Figure 7. Original image of Loop detector (normal conditions, no errors).

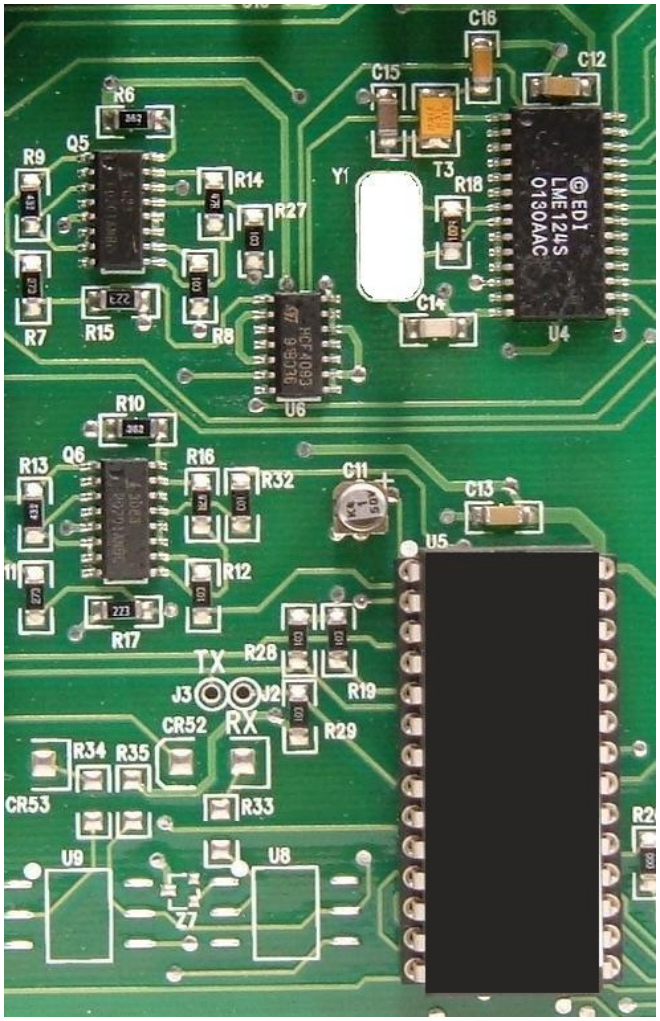

Figure 8. Same model of PCB, shown on figure 7, loop detector with a missing component Y1, white background.

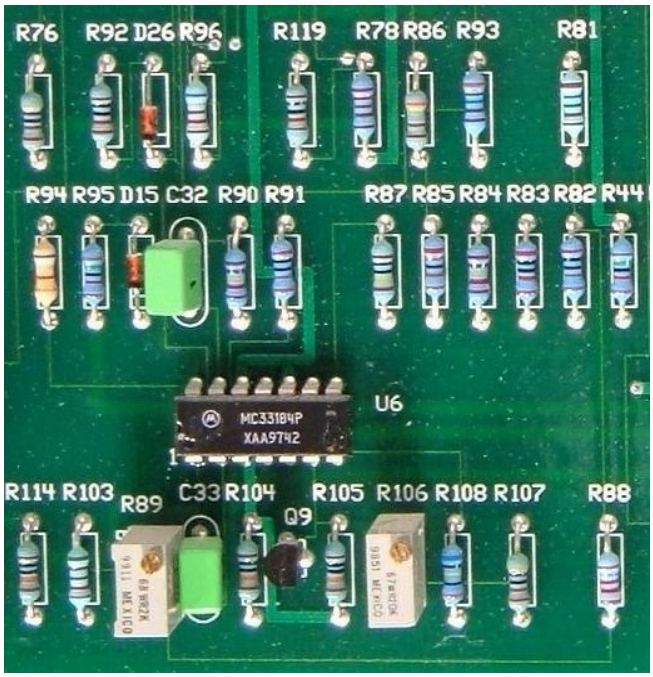

Figure 9. Original image of Modem card (normal conditions, no errors).

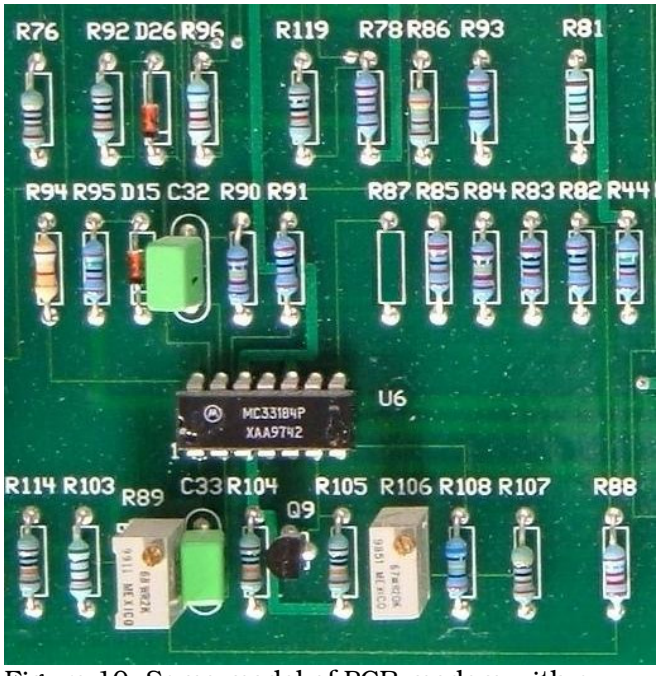

Figure 10. Same model of PCB modem with a missing component R87, green background.

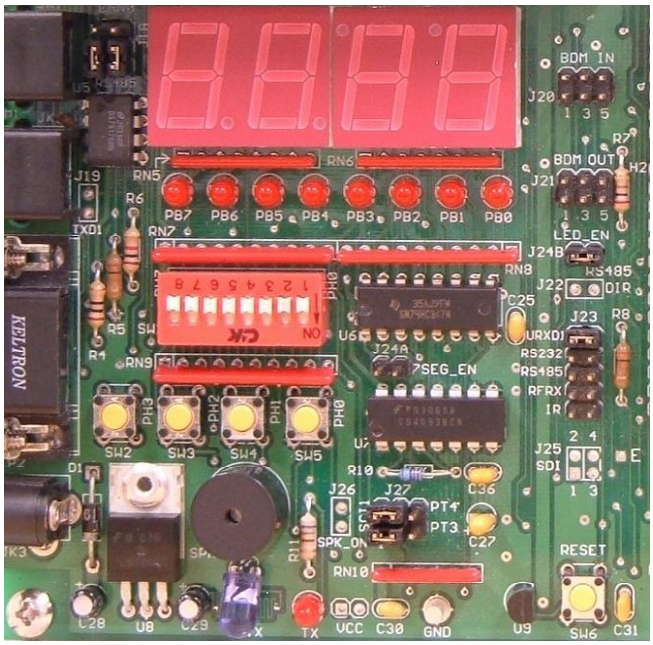

Figure 11. Original image of S12 card (normal conditions, no errors).

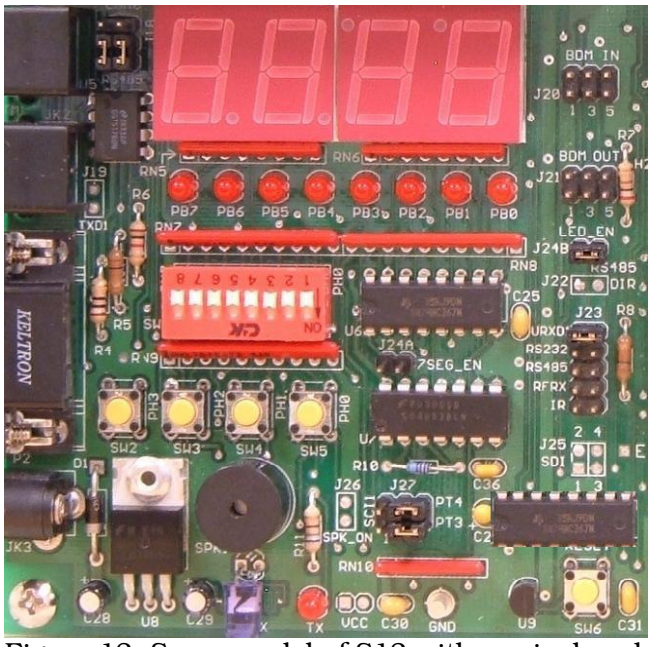

Figure 12. Same model of S12 with a misplaced component in quadrant 4 (blue background on table 3).

APPLICATION RESULTS

In order to facilitate the evaluation of the data analysis obtained from the experiment, a reference quadrant system per card type was created.

The figures 13, 14, and 15 are showing the three reference images.

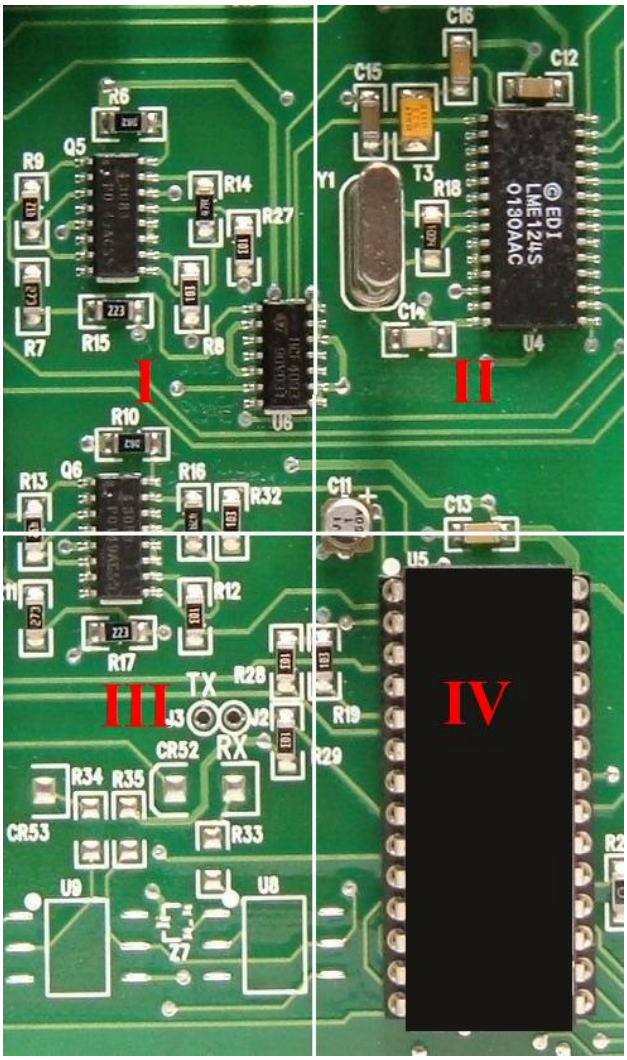

Figure 13. Reference image for a Loop Detector PCB showing the four quadrants.

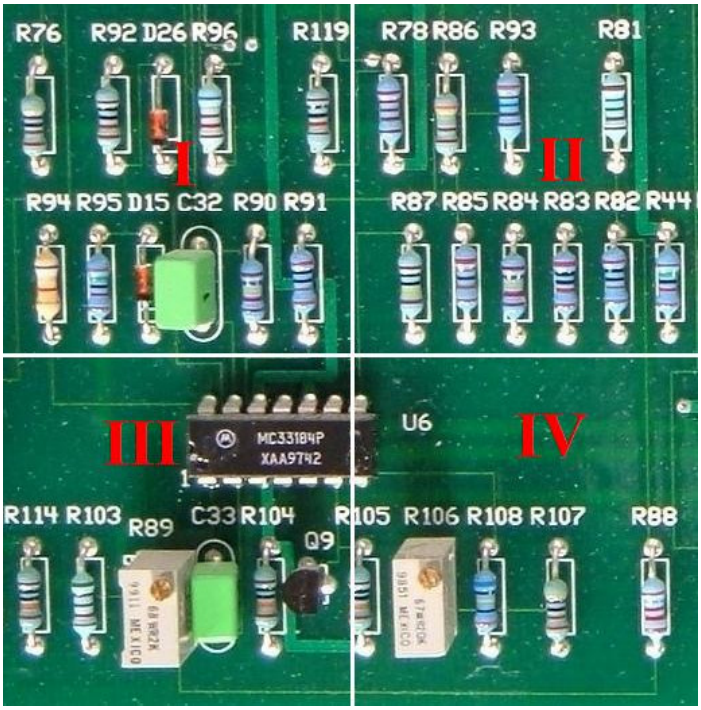

Figure 14. Reference image for a Modem PCB showing the four quadrants.

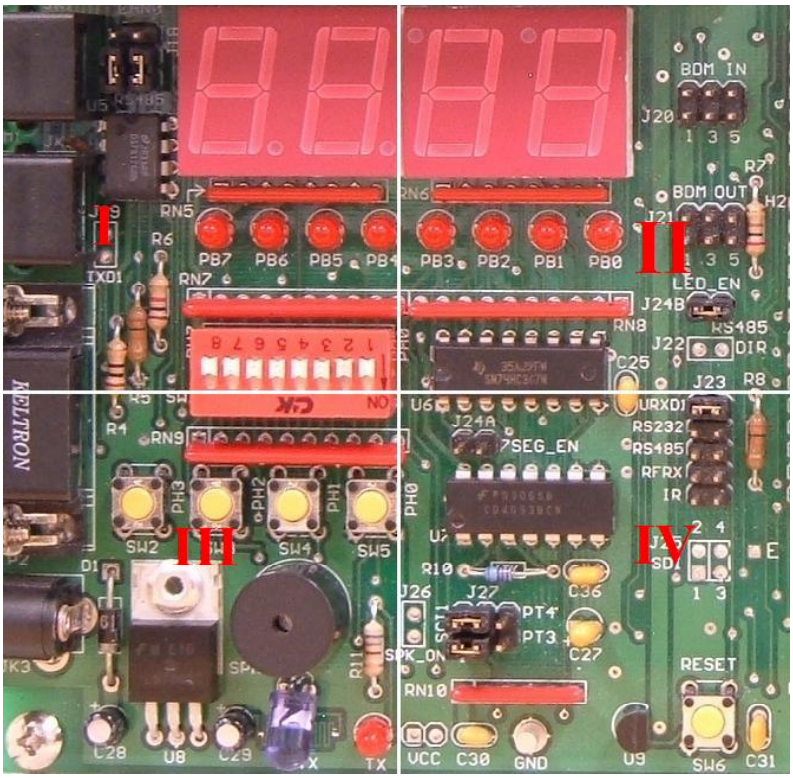

Figure 15. Reference image for a S12 PCB showing the four quadrants.

The data obtained from the experiment is displayed on the following tables.

Table 1. Data obtained from the loop detectors PCBs.

Ref	file name	Gauss 1	Gauss 2	Gauss 3	Gauss 4	Uniform	MAX	Std Dev	Comments
1.1.	loop1_err1	-2.737	-2.690	-2.090	-2.091	-1.745	2.737	0.360	Deleted U6, quadrant I
1.2.	loop1_err2	-2.368	-1.553	-1.486	-0.990	-1.797	2.368	0.571	Deleted Q5 quadrant I
1.3.	loop1_err3	-2.411	-1.595	-2.380	-1.602	-1.754	2.411	0.460	Deleted Q6, quadrant I & III
1.4.	loop1_err4	-4.379	-7.731	-2.789	-5.020	-5.918	7.731	2.060	Deleted U4, quadrant 2
1.5.	loop1_err5	-0.352	-0.351	-0.260	-0.266	-0.220	0.352	0.051	Deleted U6, quadrant I
2.1.	loop2_err1	-0.202	-0.124	-0.108	-0.063	-0.147	0.202	0.058	Deleted Q5, quadrant I
2.2.	loop2_err2	0.115	0.086	0.117	0.088	0.089	0.117	0.016	Deleted Q6, quadrant I & III
2.3.	loop2_err3	-0.602	-1.120	-0.354	-0.696	-0.854	1.120	0.319	Deleted U4, quadrant II
2.4.	loop2_err4	-15.122	-23.760	-20.711	-32.927	-24.001	32.927	7.446	Deleted U5, quadrant IV
2.5.	loop2_err5	0.643	0.515	0.395	0.321	0.471	0.643	0.141	Added U6, quadrant I
3.1.	loop3_err1	-1.945	-2.331	-1.261	-1.538	-1.467	2.331	0.469	Deleted Y1, quadrant II
3.2.	loop3_err2	0.526	0.611	0.369	0.426	0.409	0.611	0.107	Deleted Y1, quadrant II
3.3.	loop3_err3	-0.196	-0.273	-0.147	-0.214	-0.150	0.273	0.052	Added Y1, quadrant II
3.4.	loop3_err4	0.619	0.395	0.884	0.567	0.699	0.884	0.203	Added U6 , quadrant III
3.5.	loop3_err5	-4.010	-6.285	-5.559	-8.833	-6.389	8.833	2.012	Deleted U5, quadrant IV
4.1.	loop4_err1	0.943	0.751	0.589	0.476	0.676	0.943	0.203	Added U6, quadrant I
4.2.	loop4_err2	0.728	0.584	0.578	0.471	0.469	0.728	0.105	Added U6, right- quadrant I
4.3.	loop4_err3	1.290	2.005	1.049	1.646	1.360	2.005	0.418	Added U7, quadrant II
4.4.	loop4_err4	-2.485	-1.619	-2.459	-1.625	-1.805	2.485	0.491	Deleted Q6, quadrant I & III
4.5.	loop4_err5	-0.455	-0.287	-0.447	-0.286	-0.326	0.455	0.095	Deleted Q6, quadrant I & III
5.1.	loop5_err1	-0.352	-0.220	-0.203	-0.125	-0.260	0.352	0.094	Deleted Q5, quadrant I
5.2.	loop5_err2	-2.144	-1.393	-1.338	-0.879	-1.619	2.144	0.524	Deleted Q5, quadrant I
5.3.	loop5_err3	10.459	9.824	14.632	13.967	10.557	14.632	2.430	Added U5, quadrant III & IV
5.4.	loop5_err4	-2.412	-2.360	-1.839	-1.828	-1.525	2.412	0.320	Deleted U6, quadrant I
5.5.	loop5_err5	-0.448	-0.444	-0.332	-0.337	-0.272	0.448	0.064	Deleted U6, quadrant I
6.1.	loop6_err1	-4.440	-7.695	-2.820	-4.975	-5.788	7.695	2.027	Deleted U4, quadrant II
6.2.	loop6_err2	-2.180	-1.406	-2.158	-1.411	-1.593	2.180	0.439	Deleted Q6, quadrant I & III
6.3.	loop6_err3	0.787	0.613	0.730	0.577	0.518	0.787	0.098	Added Q6, of quadrant I
6.4.	loop6_err4	-0.893	-1.573	-0.553	-1.007	-1.172	1.573	0.424	Deleted U4, quadrant II
6.5.	loop6_err5	1.304	2.056	1.115	1.773	1.415	2.056	0.430	Added U4, of quadrant II
7.1.	loop7_err1	-4.081	-7.060	-2.588	-4.567	-5.365	7.060	1.859	Deleted U4, quadrant II
7.2.	loop7_err2	-2.418	-1.560	-2.396	-1.571	-1.753	2.418	0.486	Deleted Q6, quadrant I & III
7.3.	loop7_err3	-2.407	-1.549	-1.508	-0.983	-1.802	2.407	0.589	Deleted Q5, quadrant I
7.4.	loop7_err4	-0.177	-0.102	-0.095	-0.051	-0.129	0.177	0.052	Deleted Q5, quadrant I
7.5.	loop7_err5	0.275	0.463	0.246	0.415	0.356	0.463	0.106	Added U6, quadrant II & IV
8.1.	loop8_err1	0.618	0.391	0.806	0.515	0.619	0.806	0.176	Added U6, quadrant III
8.2.	loop8_err2	0.343	0.619	0.319	0.579	0.495	0.619	0.156	Added U6, quadrant II & IV
8.3.	loop8_err3	2.642	2.180	3.391	2.843	2.261	3.391	0.502	Added U4, quadrant III
8.4.	loop8_err4	0.984	0.797	0.987	0.812	0.666	0.987	0.105	Added U6, quadrant I & III
8.5.	loop8_err5	-0.446	-0.287	-0.444	-0.291	-0.318	0.446	0.090	Deleted Q6, quadrant I & III
9.1.	loop9_err1	11.355	10.371	10.075	9.352	8.010	11.355	0.830	Added U5, quadrant I, II III & IV
9.2.	loop9_err2	0.840	0.564	0.618	0.423	0.596	0.840	0.173	Added U6, quadrant I
9.3.	loop9_err3	-2.627	-1.700	-1.648	-1.079	-1.977	2.627	0.641	Deleted Q5, quadrant I
9.4.	loop9_err4	-0.222	-0.134	-0.125	-0.073	-0.165	0.222	0.062	Deleted Q5, quadrant I
9.5.	loop9_err5	0.314	0.250	0.433	0.349	0.305	0.433	0.076	Added Y1, quadrant III
10.1.	loop10_err1	1.018	0.886	1.041	0.921	0.687	1.041	0.074	Added U6, quadrant I & III
10.2.	loop10_err2	1.085	0.947	0.750	0.664	0.728	1.085	0.190	Added U6, quadrant I
10.3.	loop10_err3	-2.393	-2.470	-1.821	-1.914	-1.556	2.470	0.329	Deleted U6, quadrant I
10.4.	loop10_err4	-2.329	-1.599	-2.301	-1.607	-1.647	2.329	0.411	Deleted Q6, quadrant I & III
10.5.	loop10_err5	0.707	0.539	1.067	0.822	0.794	1.067	0.222	Added U6, quadrant III

Table 2. Data obtained from the Modem PCBs.

Ref	file name	Gauss 1	Gauss 2	Gauss 3	Gauss 4	Uniform	MAX	Std. Dev.	Comments
1.1.	modem1_err1	1.049	0.921	1.306	1.150	0.795	1.306	0.163	Deleted U6, quadrant III &IV
1.2.	modem1_err2	-0.356	-0.171	-0.212	-0.098	-0.328	0.356	0.108	Deleted R76, quadrant I
1.3.	modem1_err3	0.233	0.301	0.209	0.270	0.187	0.301	0.041	Deleted R85, quadrant II
1.4.	modem1_err4	1.141	0.829	1.025	0.745	0.716	1.141	0.181	Deleted C32, quadrant I
1.5.	modem1_err5	-0.718	-0.881	-1.194	-1.466	-1.011	1.466	0.333	Deleted R104, quadrant IV
2.1.	modem2_err1	0.291	0.335	0.256	0.295	0.188	0.335	0.033	Deleted R87, quadrant II
2.2.	modem2_err2	-0.396	-0.627	-0.341	-0.542	-0.424	0.627	0.131	Deleted R83, quadrant II
2.3.	modem2_err3	-5.987	-5.201	-7.317	-6.370	-4.479	7.317	0.879	Deleted U6, quadrant III & IV
2.4.	modem2_err4	1.547	1.028	2.641	1.748	2.058	2.641	0.672	Deleted R89, quadrant III
2.5.	modem2_err5	-1.013	-0.664	-1.723	-1.141	-1.284	1.723	0.441	Deleted R89, quadrant III
3.1.	modem3_err1	1.348	0.892	2.320	1.537	1.785	2.320	0.596	Deleted R89, quadrant III
3.2.	modem3_err2	0.109	0.113	0.187	0.194	0.141	0.194	0.046	Deleted R105, quadrant IV
3.3.	modem3_err3	-0.548	-0.570	-0.512	-0.534	-0.364	0.570	0.024	Added U6, quads I, II, III & IV
3.4.	modem3_err4	1.148	1.368	1.928	2.301	1.620	2.301	0.525	Deleted R106, quadrant IV
3.5.	modem3_err5	-0.217	-0.277	-0.368	-0.462	-0.272	0.462	0.107	Deleted R106, quadrant IV
4.1.	modem4_err1	-0.565	-0.366	-0.329	-0.209	-0.402	0.565	0.148	Deleted D26, quadrant I
4.2.	modem4_err2	0.627	0.689	0.396	0.434	0.497	0.689	0.143	Deleted R78, quadrant II
4.3.	modem4_err3	-6.686	-5.880	-8.112	-7.146	-4.992	8.112	0.932	Deleted U6, quadrant III & IV
4.4.	modem4_err4	-1.775	-2.724	-2.211	-3.387	-2.217	3.387	0.694	Added, gray , quadrant IV
4.5.	modem4_err5	0.579	0.486	0.706	0.593	0.428	0.706	0.090	Deleted U6, quadrant III & IV
5.1.	modem5_err1	0.379	0.435	0.330	0.380	0.266	0.435	0.043	Deleted R87, quadrant II
5.2.	modem5_err2	-0.537	-0.631	-0.461	-0.542	-0.367	0.631	0.070	Deleted R87, quadrant II
5.3.	modem5_err3	0.572	0.963	0.371	0.617	0.838	0.963	0.246	Added U6, quadrant II
5.4.	modem5_err4	0.767	0.594	1.251	0.957	0.915	1.251	0.281	Deleted C33, quadrant III
5.5.	modem5_err5	-0.590	-0.451	-1.044	-0.821	-0.682	1.044	0.261	Deleted C33, quadrant III
6.1.	modem6_err1	0.489	0.260	0.425	0.227	0.386	0.489	0.127	Deleted R94, quadrant I
6.2.	modem6_err2	1.584	1.052	2.723	1.810	2.103	2.723	0.697	Deleted R89, quadrant III
6.3.	modem6_err3	-0.026	-0.021	-0.047	-0.038	-0.033	0.047	0.011	Deleted Q9, quadrant III
6.4.	modem6_err4	0.448	0.509	0.391	0.445	0.331	0.509	0.048	Deleted R87, quadrant II
6.5.	modem6_err5	0.358	0.246	0.654	0.432	0.604	0.654	0.172	Deleted R89, quadrant III
7.1.	modem7_err1	1.269	0.904	1.137	0.811	0.797	1.269	0.210	Deleted C32, quadrant I
7.2.	modem7_err2	-0.511	-0.391	-0.415	-0.321	-0.296	0.511	0.079	Deleted C32, quadrant I
7.3.	modem7_err3	-0.211	-0.196	-0.277	-0.260	-0.161	0.277	0.039	Deleted U6, quadrant III & IV
7.4.	modem7_err4	-5.962	-5.227	-7.298	-6.412	-4.474	7.298	0.866	Deleted U6, quadrant III & IV
7.5.	modem7_err5	-0.942	-0.546	-1.155	-0.677	-0.794	1.155	0.272	Added, green , quadrant II
8.1.	modem8_err1	-0.724	-0.655	-0.839	-0.763	-0.503	0.839	0.077	Deleted U6, quadrant III & IV
8.2.	modem8_err2	-0.268	-0.248	-0.442	-0.409	-0.307	0.442	0.098	Deleted Q9, quadrant III
8.3.	modem8_err3	-7.049	-6.202	-8.428	-7.431	-5.156	8.428	0.923	Deleted U6, quadrant III & IV
8.4.	modem8_err4	-0.296	-0.355	-0.473	-0.573	-0.359	0.573	0.124	Deleted R106, quadrant IV
8.5.	modem8_err5	-1.058	-1.746	-1.269	-2.082	-1.389	2.082	0.462	Added, gray , quadrant IV
9.1.	modem9_err1	-0.037	-0.030	-0.064	-0.052	-0.046	0.064	0.015	Deleted Q9, quadrant III
9.2.	modem9_err2	0.421	0.384	0.367	0.335	0.259	0.421	0.036	Deleted R91, quadrant I
9.3.	modem9_err3	1.431	0.952	2.457	1.637	1.889	2.457	0.628	Deleted R89, quadrant III
9.4.	modem9_err4	-0.497	-0.450	-0.861	-0.786	-0.582	0.861	0.205	Deleted Q9, quadrant III
9.5.	modem9_err5	0.111	0.095	0.221	0.166	0.248	0.221	0.057	Deleted R89, quadrant III
10.1.	modem10_err1	-0.333	-0.201	-0.451	-0.283	-0.261	0.451	0.105	Added U6, quadrant II & III
10.2.	modem10_err2	-6.056	-5.311	-7.396	-6.502	-4.529	7.396	0.872	Deleted U6, quadrant III & IV
10.3.	modem10_err3	-2.231	-1.226	-2.480	-1.372	-1.847	2.480	0.621	Added R106, quadrant III
10.4.	modem10_err4	-0.968	-1.186	-1.639	-2.003	-1.382	2.003	0.463	Deleted R106, quadrant IV
10.5.	modem10_err5	-0.999	-1.059	-0.875	-0.930	-0.642	1.059	0.080	Added, green , quadrant II

Table 3. Data obtained from the S12 PCBs.

Ref	file name	Gauss 1	Gauss 2	Gauss 3	Gauss 4	Uniform	MAX	Std. Dev.	Comments
1.1.	S4_1_err1	-0.097	-0.113	-0.083	-0.097	-0.070	0.113	0.012	Deleted RN8, quadrant II
1.2.	S4_1_err2	0.013	0.010	0.023	0.019	0.018	0.023	0.006	Deleted purple LED, quadrant III
1.3.	S4_1_err3	-2.281	-1.885	-2.066	-1.715	-1.385	2.281	0.243	Deleted red dip switch mod, quads I & III
1.4.	S4_1_err4	-1.008	-1.219	-0.837	-1.016	-0.755	1.219	0.156	Deleted RN8, quadrant II
1.5.	S4_1_err5	-0.224	-0.179	-0.447	-0.361	-0.369	0.447	0.123	Deleted purple LED, quadrant III
2.1.	S4_2_err1	-0.419	-0.256	-0.515	-0.318	-0.338	0.515	0.114	Deleted SW2, quadrant III
2.2.	S4_2_err2	-2.068	-1.689	-3.103	-2.544	-2.040	3.103	0.611	Deleted black speaker, quadrant III
2.3.	S4_2_err3	0.582	0.381	0.788	0.518	0.518	0.788	0.169	Deleted U8, quadrant III
2.4.	S4_2_err4	-0.835	-0.544	-1.353	-0.888	-0.981	1.353	0.335	Deleted U8, quadrant III
2.5.	S4_2_err5	0.255	0.402	0.393	0.630	0.526	0.630	0.155	Added U7, quadrant IV
3.1.	S4_3_err1	1.410	1.725	0.833	1.023	1.270	1.725	0.399	Deleted right 7-segment display, quad II
3.2.	S4_3_err2	-4.496	-5.600	-2.417	-3.025	-4.496	5.600	1.439	Deleted right 7-segment display, quad II
3.3.	S4_3_err3	1.493	1.212	0.893	0.726	1.043	1.493	0.341	Deleted left 7-segment display, quad I
3.4.	S4_3_err4	-6.765	-5.552	-3.739	-3.079	-5.110	6.765	1.685	Deleted left 7-segments display, quad I
3.5.	S4_3_err5	-0.172	-0.324	-0.250	-0.462	-0.379	0.462	0.124	Added, red dip switches module, quad IV

Tables 1, 2 and 3 show the percentage change in collected power from the four Gaussians (Gauss 1, 2, 3 and 4), the Uniform filter, the Maximum change of the Gaussian power and some comments concerning the change made to the boarding. Also notice that each comment has a color background, chosen depending on the action that was taken on each PCB. For example, as can be seen on table 3, a green color means that there is a missing component that is not in its place. This left an empty space, which identifies that a component was removed (figure 10). Similarly, the blue row pertains to a component that is in the incorrect location, which is also a common PCB assembly problem (figure 12). Also, recall that the quadrants reference parts of the images to facilitate the analysis on each case, see figures 13, 14 and 15. Note that the Maximum percentage change in power value corresponds to the quadrant where any change was detected either by adding or deleting a component. For instance, on table 3 row 2.5, a U7 (black medium size IC) was added to a S12 PCB (blue

background row) on quadrant IV, and the maximum change value calculated matches the quadrant IV, Max = 0.630 = Gaussian 4.

Table 4. Data set obtained from the Matlab scripts comparing good images between them.

	Gauss 1	Gauss 2	Gauss 3	Gauss 4	uniform	Std. Dev.
loopD2.	1.655	2.038	1.497	1.834	1.942	0.236
loopD3.	-0.202	-0.450	-0.218	-0.398	-0.192	0.129
loopD4.	1.531	1.239	1.283	1.014	0.984	0.153
loopD5.	-0.831	-0.614	-0.794	-0.612	-0.556	0.104
loopD6.	1.860	1.745	1.958	1.884	2.066	0.134
loopD7.	2.511	1.530	2.539	1.703	2.021	0.443
loopD8.	1.852	1.800	2.282	2.231	1.900	0.239
loopD9.	3.368	3.211	3.065	2.909	3.250	0.155
loopD10.	-0.345	-0.603	-0.414	-0.677	-0.346	0.156
modem1.	-11.735	-11.992	-9.901	-10.209	-10.710	0.922
modem3.	-4.818	-4.832	-4.762	-4.818	-4.900	0.057
modem4.	2.975	2.905	2.791	2.660	3.048	0.165
modem5.	3.087	2.745	2.878	2.508	2.979	0.204
modem6.	-3.788	-3.834	-4.091	-4.229	-3.782	0.212
modem7.	-6.659	-7.042	-6.500	-6.896	-6.533	0.268
modem8.	-4.818	-4.832	-4.762	-4.818	-4.900	0.057
modem9.	0.634	0.638	0.448	0.380	0.701	0.152
modem10	-0.928	-1.229	-1.159	-1.457	-1.083	0.161
lab2.	-1.246	-1.123	-0.726	-0.623	-0.673	0.229
lab3.	1.565	1.453	2.054	1.941	1.605	0.281

DISCUSSION OF RESULTS

The data set shown on tables 1, 2, 3 and 4 provide adequate information to evaluate the effectiveness of the algorithm being studied. In order to help with this evaluation the following additional iterations were made:

- Calculations of standard deviation of Gaussians filter percentage power change.
- Average
- Graphical analysis

First, the calculation of the standard deviation on the images was obtained by calculating each standard deviation in each image and calculating the average value of the whole set of images of similar model. This procedure allows for the identification of images with high data dispersion between its Gaussians. As can be observed, the images on table 5 describe the average standard deviation on each set of images. The images that have a standard dev above the good images average have higher probability to be images with errors. On the other hand, images with standard deviation below good images average are images without considerable scene change with respect to original images.

Secondly, the average values for each of the sets of images for original and bad PCBs were obtained for those models that were similar. See Table 5 below.

Table 5. Average Standard Deviation of each set of images.

Image type	Average Standard Deviation (Images with Errors)	Average Standard Deviation (Good Images)
Loop Detector	0.592	0.194
Modem	0.290	0.244
S4	0.394	0.255

Third, graphical analysis provides a better view of the results. The graphics show the difference between the Gaussian powers where errors exist and those of the good images. The graphics provide a visualization of the values in the data set allowing observing how widely spread they are.

Notice that the graphics also show some peaks that correspond to images where the contrast is too high. For instance, on the set of the loop detectors, one easily can identify the highest peaks such as, on table 16, 2.4, 5.3, and 9.1. These images are correlated to the component U5, which is a device in quadrant IV, and which was added or deleted from its original place.

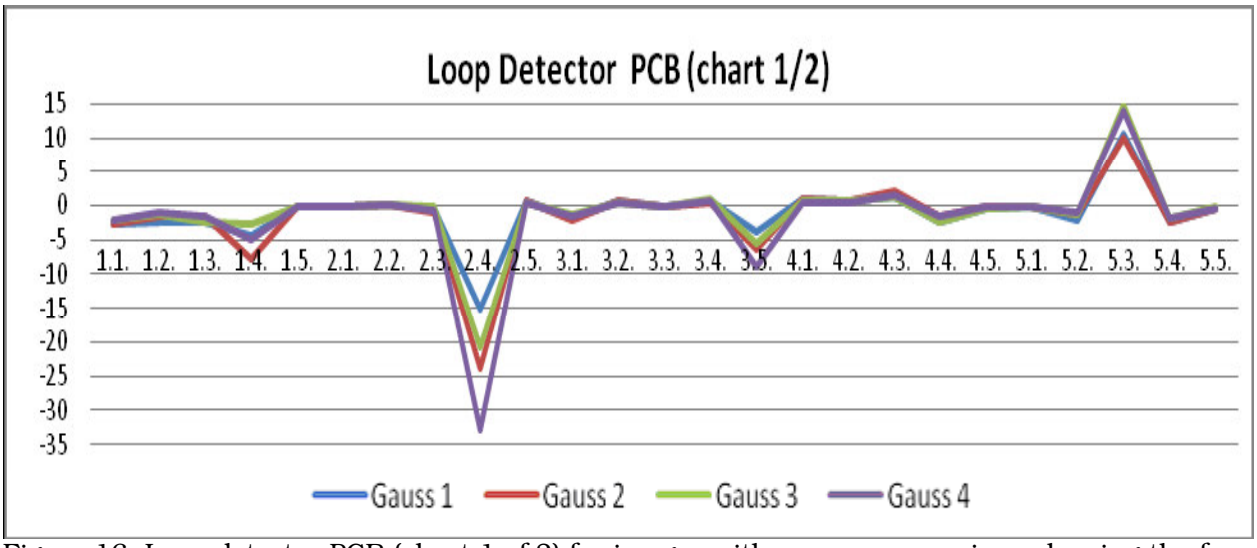

Figure 16. Loop detector PCB (chart 1 of 2) for images with errors comparison showing the four Gaussians per image.

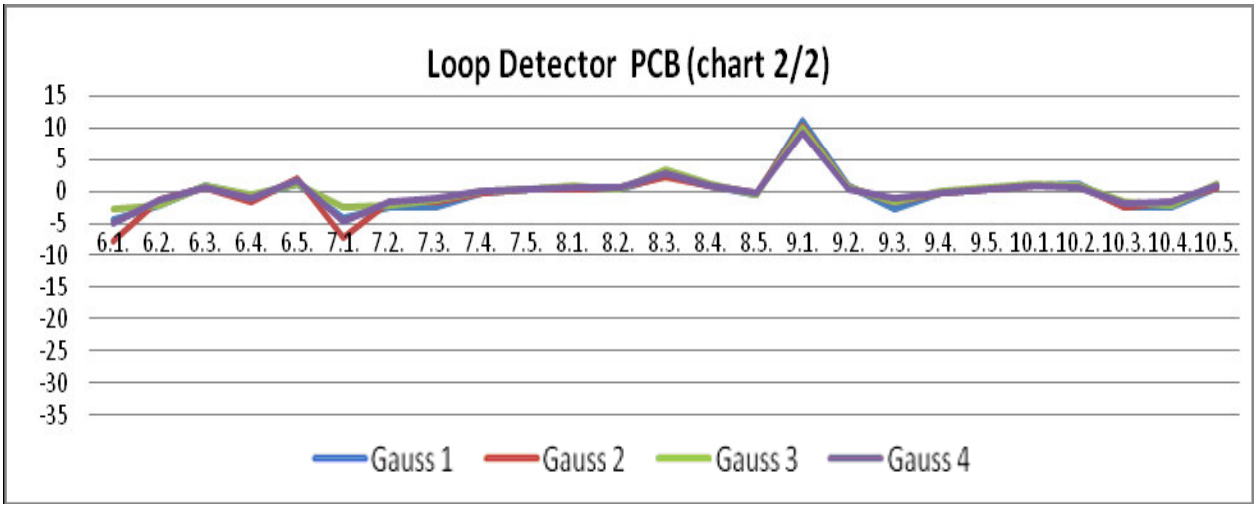

Figure 17. Loop detector PCB (cont. of figure 18) for images with errors showing the four Gaussians per image.

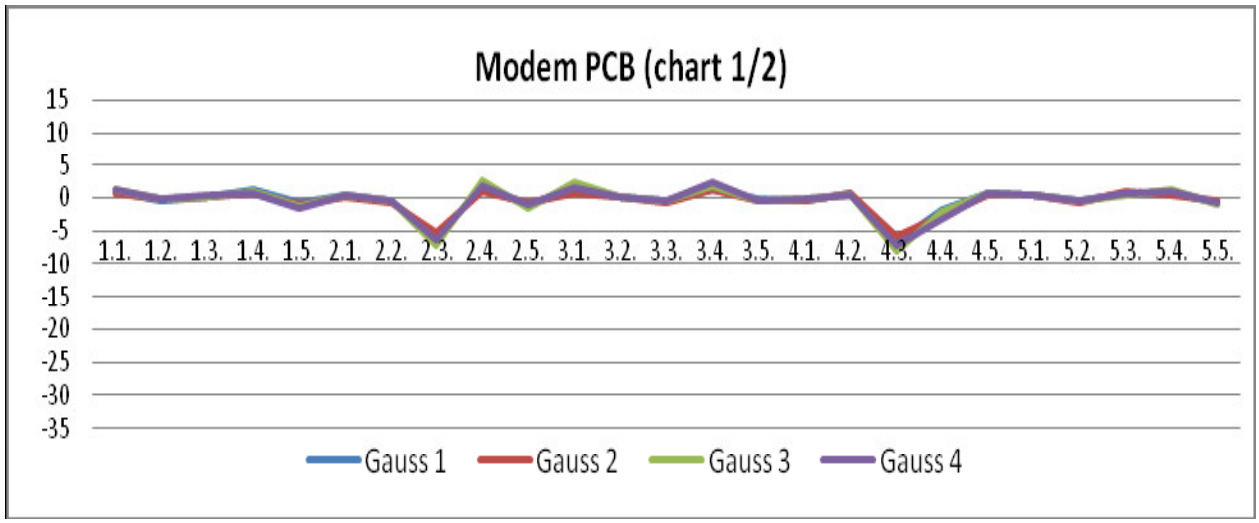

Figure 18. Modem PCB (chart 1 of 2) showing the four Gaussians per image.

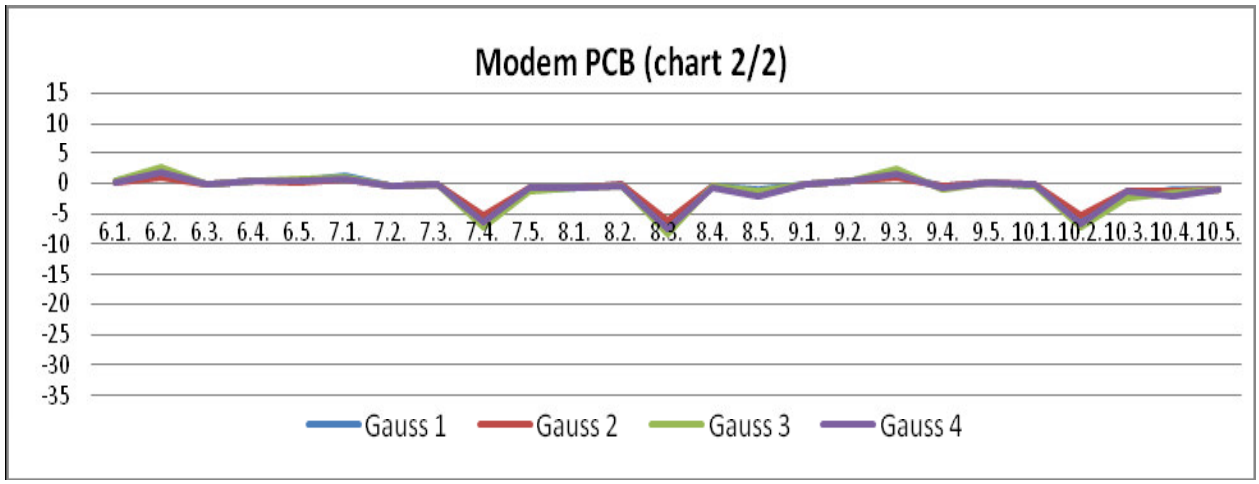

Figure 19. Modem PCB (continuation of figure 20) showing the four Gaussians per image.

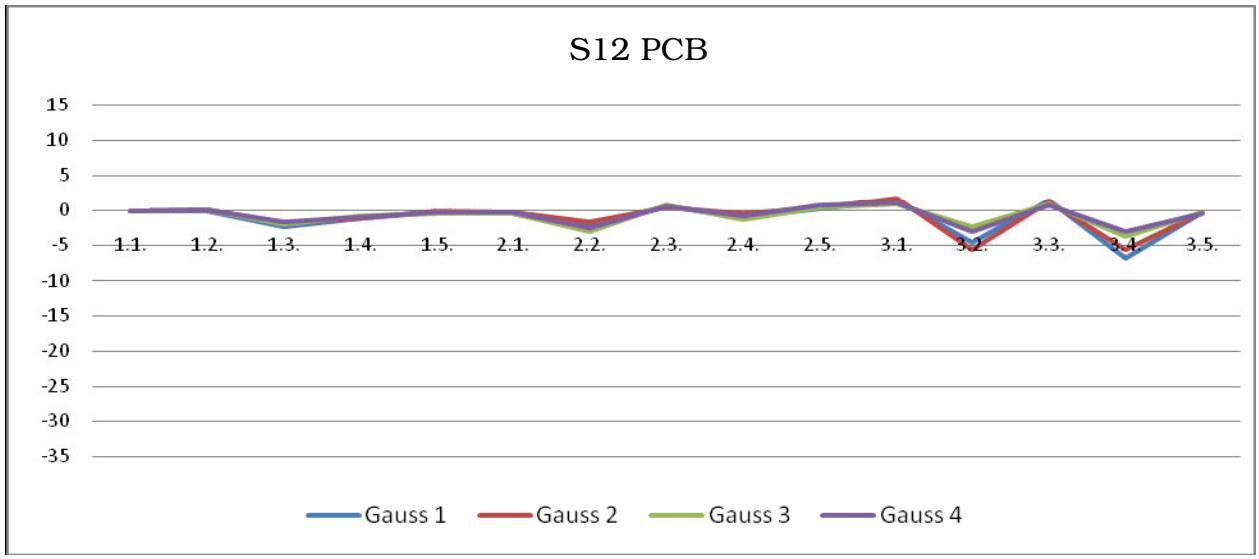

Figure 20. S12 PCB for images with errors comparison showing the four Gaussians per image.

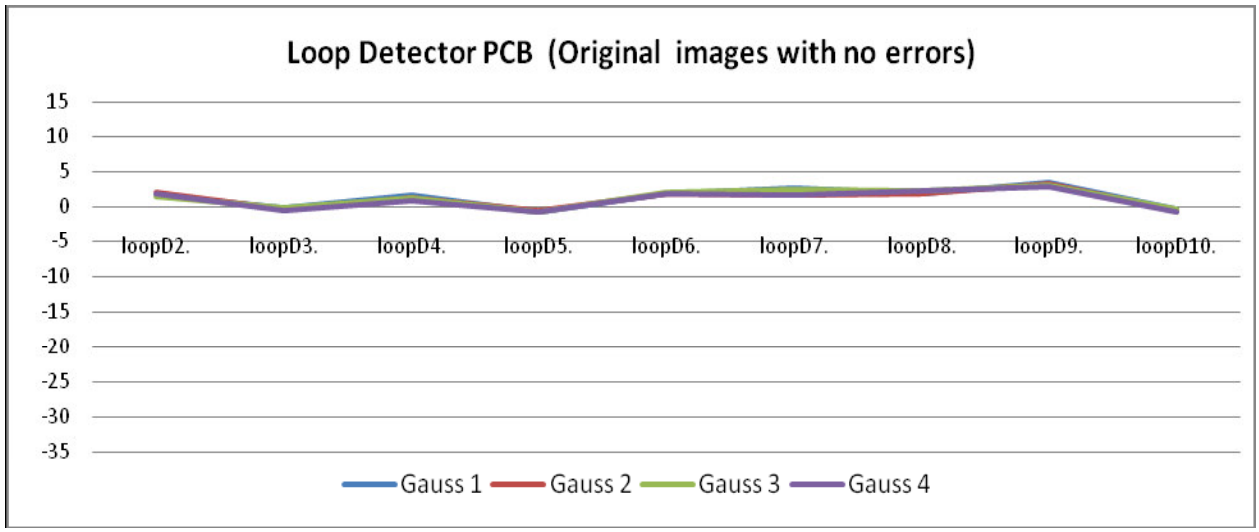

Figure 21. Loop Detector PCB for good images comparison showing the four Gaussians per image.

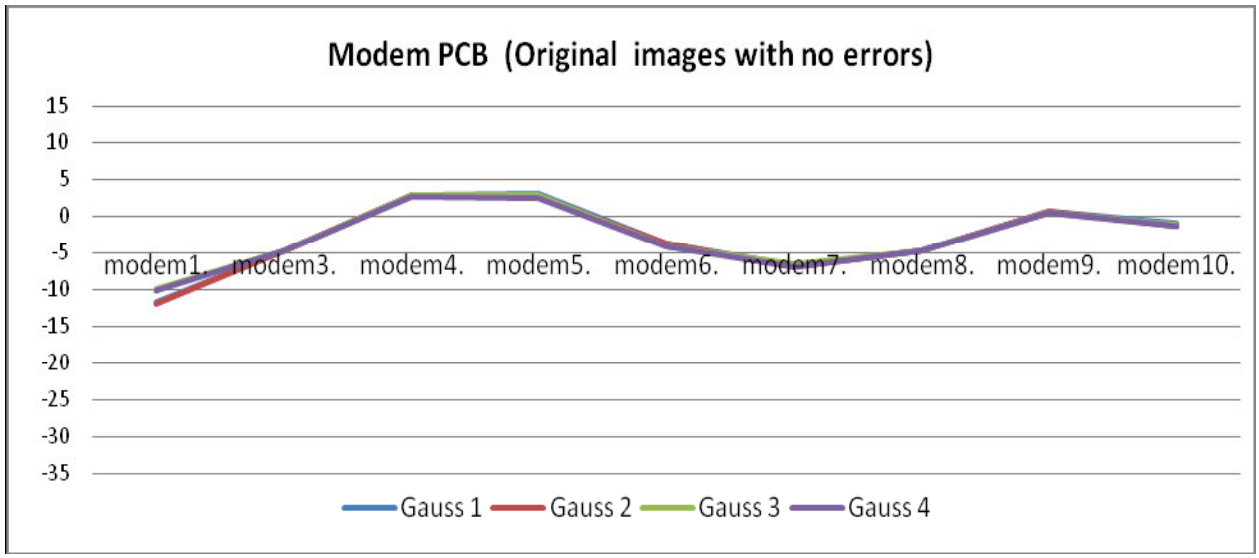

Figure 22. Modem PCB for good images comparison showing the four Gaussians per image.

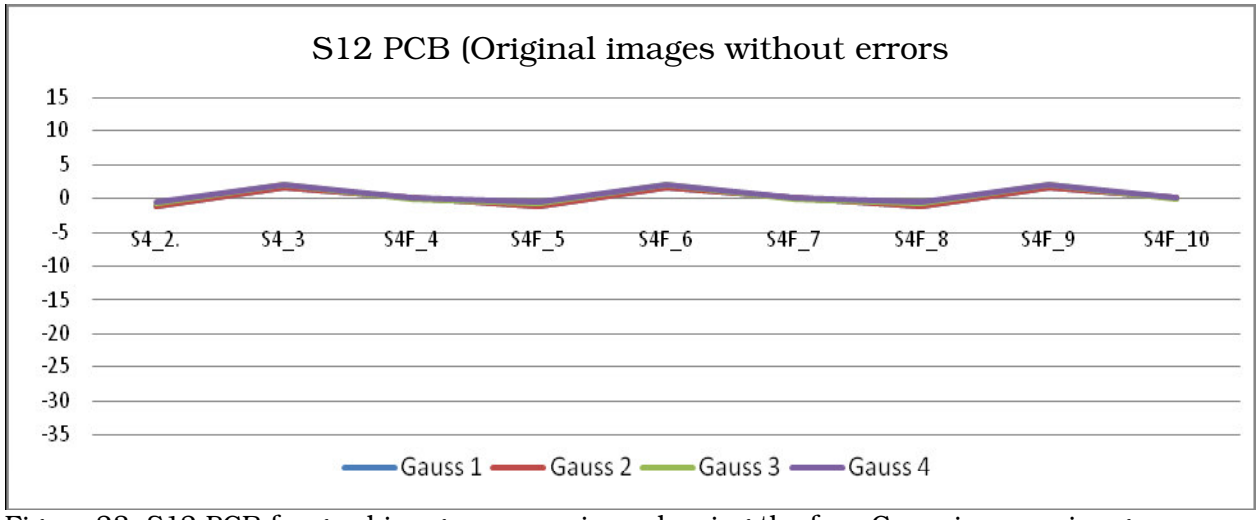

Figure 23. S12 PCB for good images comparison showing the four Gaussians per image.

TRAFFIC DETECTION APPLICATION

As previously mentioned, this algorithm is intended to be implemented also in traffic applications. This algorithm was tested under some ideal conditions (low scene change, one car). No data analysis was performed on this last step of the experiment; only visual results have been obtained. The following images show the background image used for this test and a sequence of images with the results obtained.

Figure 24. Image used as a background taken from a video detector at an intersection.

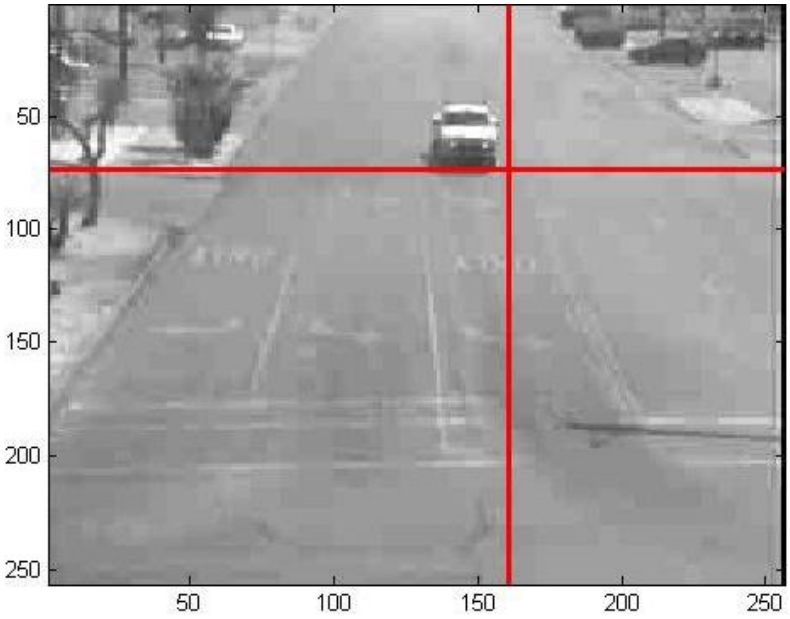

Figure 25(a)

Figure 25(b)

Figure 25(c)

Figure 25(d)

Figure 25(e)

Figure 25(f)

Figure 25 (a, b, c, d, e, f). Set of images obtained from executing the Matlab algorithm. Note that the crosshair is detecting the presence of a vehicle.

CHAPTER III

CONCLUSIONS

The data obtained from the execution of the scripts that contains the algorithm studied in this thesis supports the conclusion that the algorithm works for the detection of errors on PCBs. The algorithm has shown an acceptable level of accuracy on the error detection. The technique used on this algorithm can thus be adapted to be used on the PCB industry in order to have more accurate visual error detection. In addition, this algorithm can be adapted for industry environments utilizing video image processing tools such as Visual C/C++.

Furthermore, the algorithm has to be adapted and improved to be utilized on traffic situations. As previously mentioned, traffic situations in the field have high levels of complexity such as an excessive scene change, a high number of moving objects and a wider variety of weather conditions.

The algorithm can be improved and adapted for video processing, considering that video processing can be treated as consecutive set of pictures. In this case, a constant update of the background will be necessary.

BIBLIOGRAPHY

[1] Michalopoulos, Panos G. “Vehicle Detection Video Through Image Processing: The Autoscope System”. *IEEE Transactions on Vehicular Technology*, vol.. 40, No. 1, pp. 21-29, February 1991.

[2] Bhagat, Victor., and Wood Donald L. “Loop Detector Crosstalk”. ITE Journal, pp 36-37, 47-49. February 1997.

[3] Chatziioanoun, Alypios. Hockaday Stephen L.M. Kaighhm, Shirlee. Ponce, Leonard. “Video Image Processing Systems: Application in Transportation”. *California Polytechnic State University*, pp. 17-20, 1995.

[4] Seyfried, Robert K. “Engineering Intersections to Reduce Red-Light Running, Student notebook”, *Institute of Transportations Engineer Educational Foundation.* pp. 2, 6.

[5] Bonneson, James., and Zimmerman, Karl. “ Red-Light-Running Handbook: An Engineer’s Guide to Reducing Red-Light-Related Crashes”. *Texas Transportation Institute, The Texas A&M University System*. September 2004.

[6] Wright, Paul H. Baker, E., Jo. “Causes of Traffic Accidents”. ITE Journal, pp 41-45, 60. June 1973.

[7] Hall, Jerome. “Engineering Factors in Alcohol-Involved Traffic Accidents”. *ITE Journal*. pp 25-28. January 1986.

[8] Wordsworth, Saul. “Electronic Horizons”. *Traffic Technology International Magazine.* pp. 18-24, April/May 2008.

[9] Moya, John and Saenz, David Z. “Vehicular Traffic Monitoring Via Biologically-Inspired Approach”. *WSEAS Transactions on Signal Processing, vol. 1, issue 3, pp. 385-391, December 2005.*

[10] Moya, John and Saenz, David Z. “Just-Enough-Smart Sensing Aplied to Printed Circuit Board Inspection”. *Paper on review.*

[11] Horng-hai Loh and Ming-Sing Lu, “Printed circuit board inspection using image analysis”, *IEEE Trans. Ind. Applicat.*, vol. 35, pp. 426-432, March 1999.

[12] Saenz, David Z. Dissertation. *University of Texas at El Paso*, 2005.

[13] Yin, Jun. Li, Dong-Guang. Fang, Hui-Min. Liu, Zhi-Feng. “Moving Objects Detection by Imitating Giologic Vision Based on Fly’s Eyes”. *Proceedings of the 2004 IEEE Int. Conf. on Robotics and Biomimetics*, Shenyang, China, August 2004.

[14] Moya, J.A., Donohoe, G. W., Wilcox, M.J. “Cellular Modeling of a Fly Photoreceptor”. *Depts. Of Electrical and Computer Engineering and Psychology*, University of New Mexico, Albuquerque, NM. 1997.

[15] Moya, J.A., Donohoe, G. W., Wilcox, M.J. “A Model of the Neuro-ommatidium of the Fly’s Eye*”. Depts. Of Electrical and Computer Engineering and Psychology*, University of New Mexico, Albuquerque, NM. 1997.

[16] Völkel, R., Eisner, M., and Weible, K. J. 2003. "Miniaturized imaging systems." *Microelectron. Eng.* 67-68, 1 (Jun. 2003), 461-472.

CURRICULUM VITAE

Daniel Flores was born in El Paso, Texas in 1973. The fifth son of Jesus T. Flores and Rosalía Rodríguez Santos, he graduated from *Centro de Bachillerato Tecnológico Industrial y de Servicios # 128* (High School), Ciudad Juarez, Chihuahua, México, in the spring of 1991 earning the High School diploma and degree of Electronic Technician and entered *Instituto Technológico de Ciudad Juárez* (Juarez Institute of Technology) in the fall of 1991. While pursuing a bachelor's degree in Electronics Engineering, he worked as an Electronic Technician for the Test Engineering Department of Harman Consumer Manufacturing, a consumer electronics company in El Paso, Texas during February 1997 and January 1998. After receiving his bachelor's degree in Electronics Engineering, he worked for the City of Juarez (Municipio de Ciudad Juarez, Chihuahua) as Traffic Management Center Coordinator from February 1998 to February 2001, moving to the City of El Paso Traffic Engineering Department as a Traffic Engineering Associate where he is currently working. In the spring of 2004, he entered the Graduate School at University of Texas at El Paso, Department of Electrical and Computer Engineering.

Permanent Address: 1544 Cherokee Ridge

El Paso, Texas 79912