ISSN : 2210-142X

International Journal of Computing and Digital Systems (IJCDS)

Volume No. 13
Issue No. 1
January - April 2024

ENRICHED PUBLICATIONS PVT. LTD

JE-18, Gupta Colony, Khirki, Extn, Malviya Nagar, New Delhi-110017
PHONE : - + 91-8877340707
E-Mail : info@enrichedpublications.com

ISSN : 2210-142X

International Journal of Computing and Digital Systems (IJCDS)

Aims and Scope

International Journal of Computing and Digital Systems (IJCDS) is a peer-reviewed International Journal that currently publishes 6 issues annually. IJCDS journal publishes technical papers, as well as review articles and surveys, describing recent research and development work that covers all areas of computer science, information systems, and computer / electrical engineering.

The topics covered by IJCDS are including and not limited to the following research areas:

- Reconfigurable Computing & Embedded systems
- Computer Communications and Networking
- Internet of Things & Real Time Systems
- Cyber Security
- Cloud Computing
- Smart Systems
- Information Systems and Communication Service
- Innovation/Technology Management
- Business Information Systems
- Software Engineering
- Mobile & Web Applications
- Theory of Computation
- Data Structures, Cryptology and Information Theory
- Artificial Intelligence & Robotics
- Image Processing, Computer Vision, Pattern Recognition & Graphics
- Data Mining & Big Data
- Smart Grids & Renewable Energy
- Human Computer Interaction

Responsiveness Speed

Time from submission to first decision after peer review: 4-8 Weeks
Time to immediate reject: 2-4 Weeks

Publication Speed

Time from final acceptance to print: 3-6 Months
Total time to publication: 4-8 Months
Time from submission to acceptance: 4-12 Weeks
Time from acceptance to online Publishing: 4-8 Weeks

Publishing Ethics

Plagiarism in all its forms constitutes unethical publishing behavior and is unacceptable. The journal has no tolerance on plagiarism. All submitted manuscripts must go through cross checking using turnitin as an online plagiarism checker.

ISSN : 2210-142X

International Journal of Computing and Digital Systems (IJCDS)

(Volume No. 13, Issue No. 1, January - April 2024)

Contents

ISSN 2210-142X

A Portable and Low Cost System to Blood Glucose, Cholesterol and Urea Identification

José Renato Garcia Braga1, Alvaro Antonio Alencar Queiroz2 and Alexandre Carlos Brandão Ramos3

1Federal University of Itajubá: Mathematic and Computing Department, Itajubá, Brazil
2Federal University of Itajubá: Physics and Chemistry Department, Itajubá, Brazil
3Federal University of Itajubá: Mathematic and Computing Department, Itajubá, Brazil

ABSTRACT

Over the last century there has been a considerable increase in human longevity and this made a large number of people to reach a critical age for development of several diseases. As a result of this increase in life expectancy health issues related, some examples are hypercholesterolemia, hyperglycemia and increased levels of blood urea. This paper presents a portable and low cost system using Artificial Neural Networks to blood metabolites identification. The system developed is based in amperometric biosensors and is able to perform the identification of glucose, cholesterol and urea concentrations in the blood. The main goals of this system is: the identification of three types of blood metabolites with their concentrations, the low cost of the entire system and the reuse capability of the biosensor.

Keywords: artificial neural networks, chemical and biological sensors, monitoring blood metabolites

1. INTRODUCTION

Hypercholesterolemia, which is the presence of high levels of cholesterol in the blood, is not a disease but a serious metabolic derangement due to the fact that cholesterol has an important role in the pathogenesis of atherosclerosis. Hyperglycemia, characterized by excess of sugar in the blood, is a disorder that can cause several complications such as amputation and blindness. Too much urea in the blood is a strong indicator that the individual may have liver and kidney diseases.

Having these problems in sight make it is easy to notice how important the control of glucose, cholesterol and blood urea levels is. Then comes the need to create a device to monitor the levels of these blood metabolites simultaneously and daily once they can rapidly change.

Other than being a non-invasive tool, the biosensors devices provide in the health system a considerable saving compared to conventional methods. Another advantage of using biosensors for monitoring blood metabolites is due to the fact that there are patients who require daily monitoring, for example, diabetic patients.

Biosensors are electronic devices capable of converting a biological reaction in an appropriate signal. This signal can be potentiometric, amperometric, conductimetric, optical, piezoelectric or enthalpymetric. Figure 1 shows the basic components of a biosensor. In (a), we have the biocatalyst, where the biochemical reaction responsible for generating the signal occurs in (b) we have the transducer in (c) the amplifier and in (d) the results.

The clinical analysis methods that are currently being used in laboratories are expensive since they rely on duly qualified staff, large instruments, and appropriated place among other costs. Also, these tests bring some discomfort to the patient because of the need for a significant amount of biological fluid, which is usually collected with the use of needles and syringes, and thus can be considered invasive methods.

2. BIOSENSORS

The biosensors then emerge as an alternative to replace conventional methods from clinical laboratories. The analysis can be done by the patient himself at any time and place with the need of only a single drop of blood.

Other than being a non-invasive tool, the biosensors devices provide in the health system a considerable saving compared to conventional methods. Another advantage of using biosensors for monitoring blood metabolites is due to the fact that there are patients who require daily monitoring, for example, diabetic patients.

Biosensors are electronic devices capable of converting a biological reaction in an appropriate signal. This signal can be potentiometric, amperometric, conductimetric, optical, piezoelectric or enthalpymetric.

Figure 1 shows the basic components of a biosensor. In (a), we have the biocatalyst, where the biochemical reaction responsible for generating the signal occurs in (b) we have the transducer in (c) the amplifier and in (d) the results.

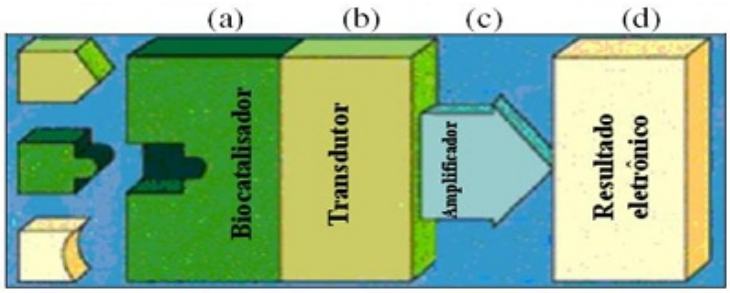


Figure 1. Representation of a biosensor: biocatalyst.

The first biosensor was developed by Clark and Lions in 1962, and this became known as enzymatic electrodes, it can be seen in the figure below.

The enzymatic electrodes or biosensors are used in a number of analytical determinations, in which the detection rate of glucose on the blood is highlighted. From the invention of Clark and Lyons, several types of biosensors have been developed for different types of clinical analysis, obtaining the most prominent biomedical analysis such as the monitoring of hemometabolite in human blood (such as glucose, cholesterol and urea).

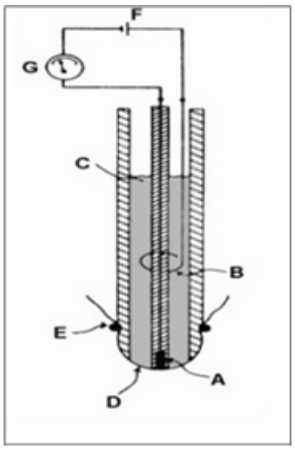


Figure 2 – Enzymatic biosensor by Clark e Lyons.

The biosensors are usually classified according to the signal that is generated in the transducer. The amperometric biosensors are the most used and researched, as it can be seen in Figure 3. But there are other types of biosensors based on other types of signals.

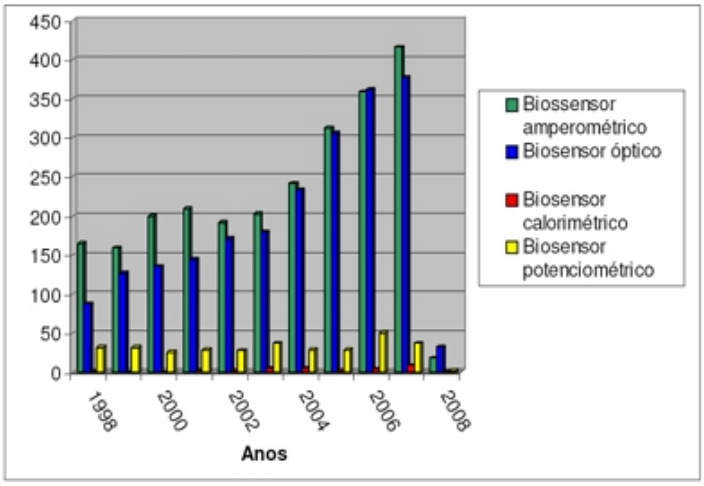


Figure 3 – Comparison chart for the most studied types of biosensors.

The amperometric biosensors provide the ability to capture an electrical signal that is proportional to the analyte concentration (hemometabolite analyzed). They are the most common biosensors found in the market and are shown in Figure 4.

Figure 4 – Commercial amperometric biosensors used in clinical analysis.

3. THE CHEMISTRY IN BIOSENSORS

The main components of a biosensor are the enzymes. The enzymes are biocatalysts that have high selectivity rate. They are proteins with the specific function of speed up chemical reactions that occur under unfavorable thermodynamically conditions. They considerably accelerate the speed of chemical reactions in biological systems when compared to corresponding non-catalyzed reactions. This is achieved by lowering the activation energy required for a chemical reaction, resulting in increased speed in the reaction and enabling the metabolism of living beings. The catalytic ability of enzymes makes them suitable for industrial applications such as in pharmaceuticals or food industry. The enzyme acts on the substrate witch is transformed into a product. In the absence of enzyme little product is formed, otherwise, the reaction is processed at high speed. Enzymes are the most specific catalysts known for both the substrate and for the reaction performed on the substrate.

3.1 Immobilization

In a biosensor the enzyme is usually immobilized on the electrode surface along with another reagent that will react with one of the blood metabolites. In this study, the enzymes are immobilized using dendrimers.

Dendrimers are monodisperses macromolecules, highly branched, with well-defined structures and uniform molecular weight, as shown in Figure 5. This class of compounds has received great attention from researchers because they have molecular uniformity, multifunctional surface and the presence of internal cavities. These specific properties make dendrimers suitable for the immobilization of enzymes.

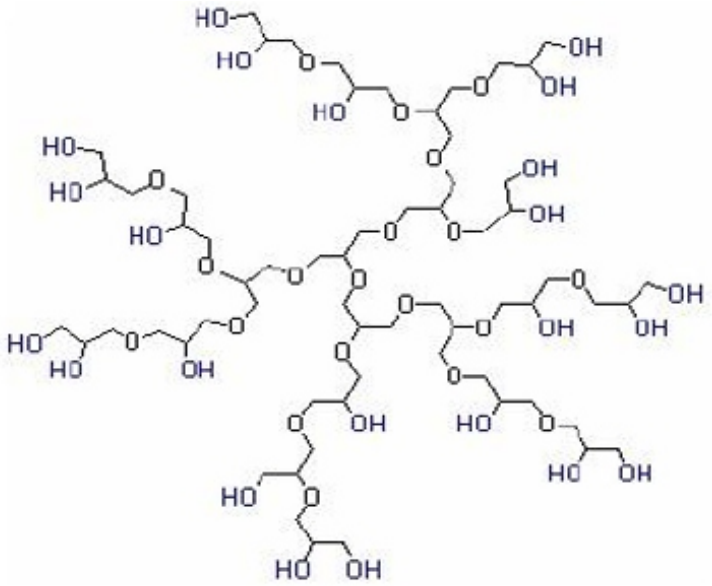


Figure 5 – Illustration of a dendrimer structure.

4. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks are computational techniques presented in a mathematical model inspired by the neural structure of intelligent organisms and acquiring knowledge through experience, Figure 6 shows the structure of a neuron.

An artificial neural network can have thousands of processing units. Furthermore the brain can have billions of neurons.

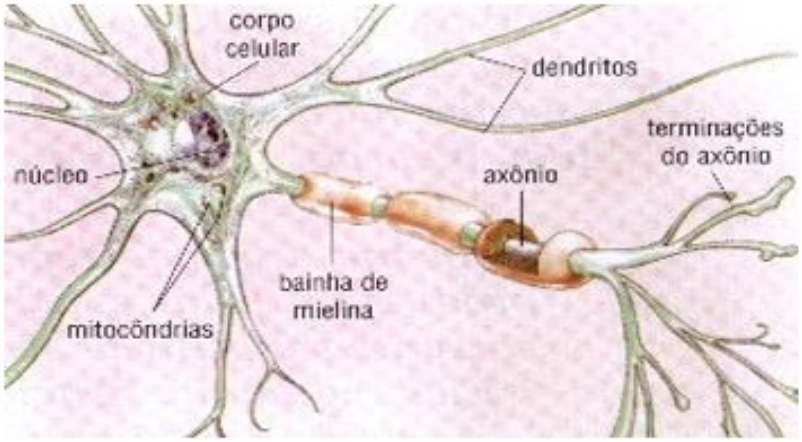


Figure 6 – Delineation of the structure of the constituents of a biological neuron.

The nervous system consists of an extremely complex set of neurons. The neuron can be considered the basic unit of the structure of the brain and the nervous system. The communication is done through pulses, when a pulse is received the neuron processes it, and after a certain action threshold is reached, the neuron triggers a second pulse that produces a neurotransmitter substance which flows from the cell body to the axon. They have an essential role in determining the performance, behavior and ratiocination of human beings.

4.1 General features

The artificial neural models, try to bring the computer processing to the human brain, and they correspond to algorithms that mimic the biochemical process of the brain and are similar to this on two points: a) Knowledge is gained through learning steps and b) Synaptic weights are used to store knowledge. Synapse is the name given to the connection between neurons, in it is assigned values that are called synaptic weights. The artificial neuron is a binary logical-mathematical device that tries to simulate the functions and behavior of a biological neuron. Thus, dendrites are represented by entries where connections with the artificial cell body are made through elements called weights that simulate synapses.

Stimuli captured by the input are processed by the weighted sum function of signals, and the threshold for triggering the biological neuron was replaced by the transfer function. Combining several artificial neurons we can form the socalled Artificial Neural Network, a schematic example is shown in Figure 7. From an artificial neural network formed, a series of values can be applied to a neuron, and this is connected to others by the network. These values (or entries) are multiplied in the neuron by the weight value of their synapses. Then these values are added.

If this sum exceeds an established limit value, a signal is propagated through the output (axon) of this neuron. Then, this same step is performed by other neurons in the network. This means that neurons will face some kind of activation, depending on the inputs and the synaptic weights. There are several ways to develop a neural network. It must be mounted according to the problems to be solved. In its architecture is determined the number of layers used (the layers are formed by neurons) and the number of neurons in each layer, the type of synapse used.

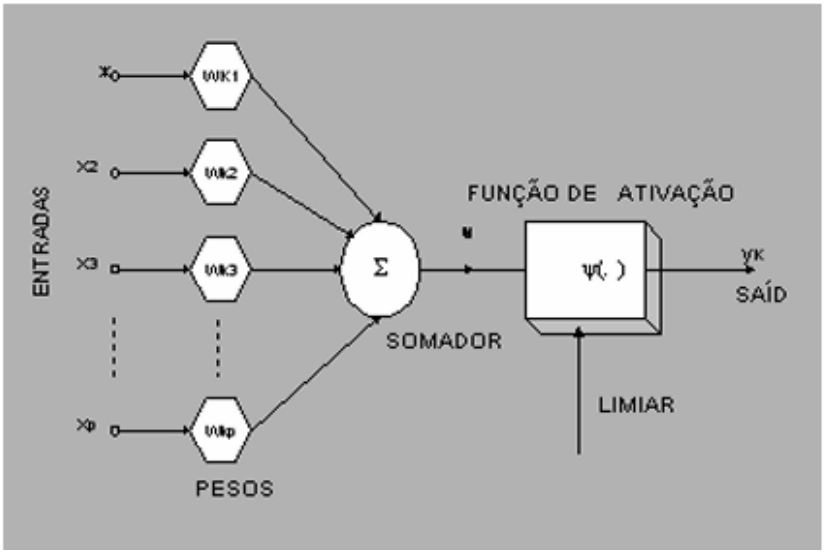


Figure 7 – Representation of the architecture of an artificial neural network (ANN).

4.2 Training

The most important property of neural networks is the ability to learn and thereby improve their performance. They learn by rote memorization, contact, examples, by analogy, by exploration and also by discovery. This is done through training, iterative process of adjustments applied to the weights. Learning occurs when the artificial neural network reaches a generalized solution to a class of problems. It is called learning algorithm a well-defined set of rules for solving a learning problem. There are many types of learning algorithms specific to certain models of ANNs (Artificial Neural Networks) they differ from each other by how the weights are modified.

The learning process of neural networks is performed when there are several significant changes in the synapses of neurons. These changes occur according to the activation of neurons. If some connections are used more frequently these are enhanced while others are weakened.

For this reason when an artificial neural network is deployed for a given application it takes time for it to be trained. There are basically three types of learning in artificial neural networks:

(I) Supervised; in this type, the neural network receives a standardized set of inputs and corresponding output patterns, there are adjustments in synaptic weights until the error between the output patterns generated by the network has a desired value;

(ii) Non-supervised; in this type, the neural network process the data to determine some properties of the data set. From these properties the learning is made;

(iii) Hybrid; this type is a merge of supervised and unsupervised. Thus, a layer can work as one type while the other layer works with the other type.

4.3 Multilayer perceptron

The artificial neural network is a system of neurons connected by synaptic connections and divided into input neurons that receive stimuli from the external environment, internal or hidden neurons and output neurons, which communicates with the exterior. The way to arrange the neurons into distinct layers is called a multilayer perceptron. The multilayer perceptron, as the diagram in Figure 8, was designed to solve complex problems, which could not be resolved by the basic model of neural network, also known as single-layer perceptron, witch only has the input and output layer of neurons.

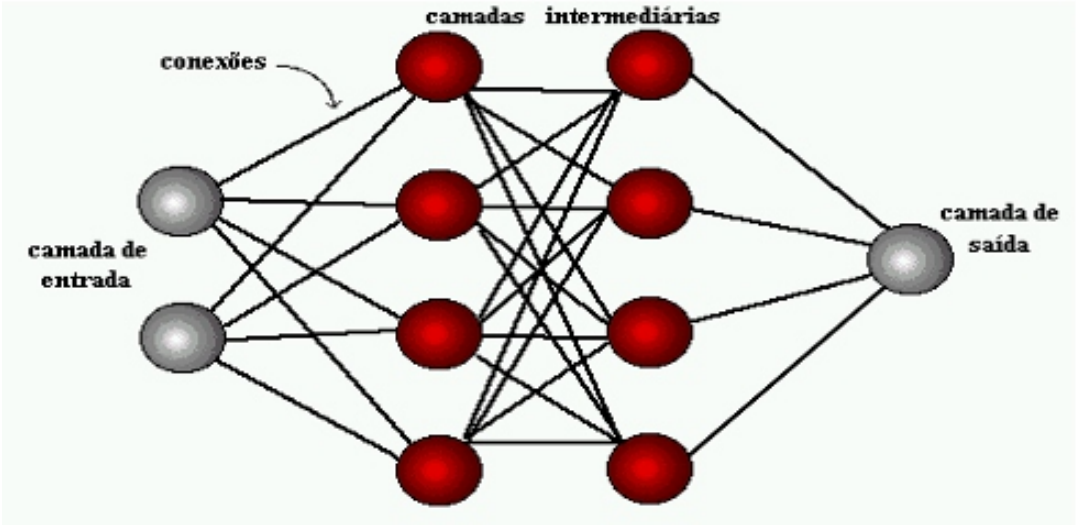


Figure 8 – Representation of a multilayer perceptron.

The internal neurons are of paramount importance for the neural network once it has been proven that without them it becomes impossible to solve problems not linearly separable. In other words we can say that a network is composed of several processing units, whose operation is quite simple. Usually the layers are classified into three groups: - Input Layer: where patterns are presented to the network; - Intermediate or Hidden Layers: where most of the processing is done through weighted connections, can be considered as features extractors; - Layer Output: where the final result is completed and submitted.

4.4 The backpropagation algorithm

The backpropagation algorithm uses supervised learning this means he try to find iteratively the smallest difference between the desired outputs and the outputs obtained by the artificial neural network, with a minimum error. It works by adjusting the weights between the layers through the backpropagation by correcting the error detected, with other words, propagating the error from the output layer to the input layer in each iteration. Networks using backpropagation work with the generalized delta rule suitable for multilayer networks. The default delta rule essentially program a descendent gradient in the squared sum of the error for linear activation functions, logical threshold function. Network without intermediate layers can solve problems where the error surface has the shape of a paraboloid with only a minimum. In these cases we should use a network with intermediate layers. Still, the networks are subjected to problems with local minima. The generalized delta rule, which will be discussed later, works when used on the network units with a semi-linear activation function, which is a continuous differentiable and non decreasing function. An activation function widely used in these cases is the sigmoid function.

The learning rate is a proportionality constant because this learning procedure requires only that the change in weight be proportional to η. However, the true gradient requires that infinitesimal steps be taken.

So the larger this constant the greater the change in weights, increasing the speed of learning, which can lead to a model oscillation on the error surface. The ideal would be to use the higher learning rate as possible that does not lead to an oscillation, resulting in a faster learning.

The training of MLP (Multilayer Perceptron) networks with backpropagation may require many steps in the training set, resulting in a considerably long time of training. If a local minimum is found the error for the training set stop its decrease and stay in highly acceptable value. One way to increase the learning rate without increasing the oscillation is to modify the generalized delta rule to include the term momentum, a constant that determines the effect of past changes in the weights of the current direction of movement in space of weights. This way the momentum term takes into account the effect of previous weight changes in the direction of the current movement in the space of weights. The momentum term becomes useful in error spaces containing long throat, with sharp curves and valleys with gentle descents.

Once the network is trained and the error is in a satisfactory level it can be used as a tool for classification of new data. For this, the network should be used only in feedforward mode. In other words, new entries are presented to the input layer then processed in the intermediate layers and the results are presented in the output layer, as in training, but without the backpropagation. In addition, you may need to preprocess the data, through standardization, escalations and format conversions to make them more appropriate for the network usage. In the interpretation of the network the output shown is the data model.

The ANNs using backpropagation as well as other types of ANNs can be seen as "black boxes" in which almost no one knows why the network reaches a certain result, where the models do not have explanations for the answers. In this regard many studies are performed to extract knowledge from ANNs and the creation of explanatory procedures, which in certain situations can justify the conduct of ANNs.

Another limitation is the training time of ANN (Artificial Neural Network) using backpropagation, which tends to be pretty slow. Sometimes it takes hundreds of cycles to reach an acceptable level of error, especially if it is being simulated in ordinary computers, because the CPU (Central Processing Unit) must calculate the functions for each unit and its connections separately, which can be problematic in very large networks or with large amounts of data. It is very hard to define the optimal architecture of ANNs so that it is as large as necessary to obtain the necessary representations, while small enough to have a faster training.

There are no clear rules to define how many artificial neurons an intermediate layer must possess, how many layers there should exist, not even how each connection between units should be.

4.5 ANN development

4.5.1 Data Collection

The first step of development of an ANN is the collection of data of the problem and the separation into a training set and test set. This task requires a careful analysis of the problem in order to guarantee no ambiguities or data errors. Furthermore, the data must be meaningful and cover the entire problem domain and should not cover only the normal or routine operations but also the exceptions and the boundary conditions of the problem domain.

Typically all the data collected are separated into two categories: training data which are used for training the ANN and test data which will be used to verify the performance of ANN under real conditions of use. Besides this division, you can also use a subset of the training set, creating a validation set, used to verify the efficiency of ANN and also as a stopping criterion of the training. After determining the sets they are placed in random order for prevention tendencies associated with the order of presentation of data. In addition, you may need to pre-process the data through standardization, escalations and format conversions to make them more appropriate.

4.5.2 ANN Configuration

After collecting the data it is time to setup the configuration of the ANN, which can be divided into the following steps:

I. Selection of an appropriate neural paradigm for the application.

II. Topology determination of the ANN to be used (number of layers, number of artificial neurons in each layer, etc.).

III. Determination of parameters of the learning algorithm and activation functions. This step has a huge impact on the system performance. Usually these choices are made on an experimental basis. The setting of ANNs requires experience designers.

4.5.3 ANN Training

At this stage following the training algorithm chosen, as shown in Figure 9 the weights of the connections are adjusted. It is important to consider aspects such as the initialization of the ANN, the training mode and the time spent to train it. A good choice for weights initial values can significantly reduce the training time. Normally, the weights initial values of the network are random numbers and distributed uniformly in a defined range. The wrong choice of weights can lead to a premature saturation. As to training, in practice the most widely used is the default mode due to the smaller data storage, in addition to being less susceptible to the problem of local minima. On the other hand, in batch mode if you have a better estimate of the gradient vector, which makes training more stable. The relative efficiency of the two training modes depends on the problem that is the case.

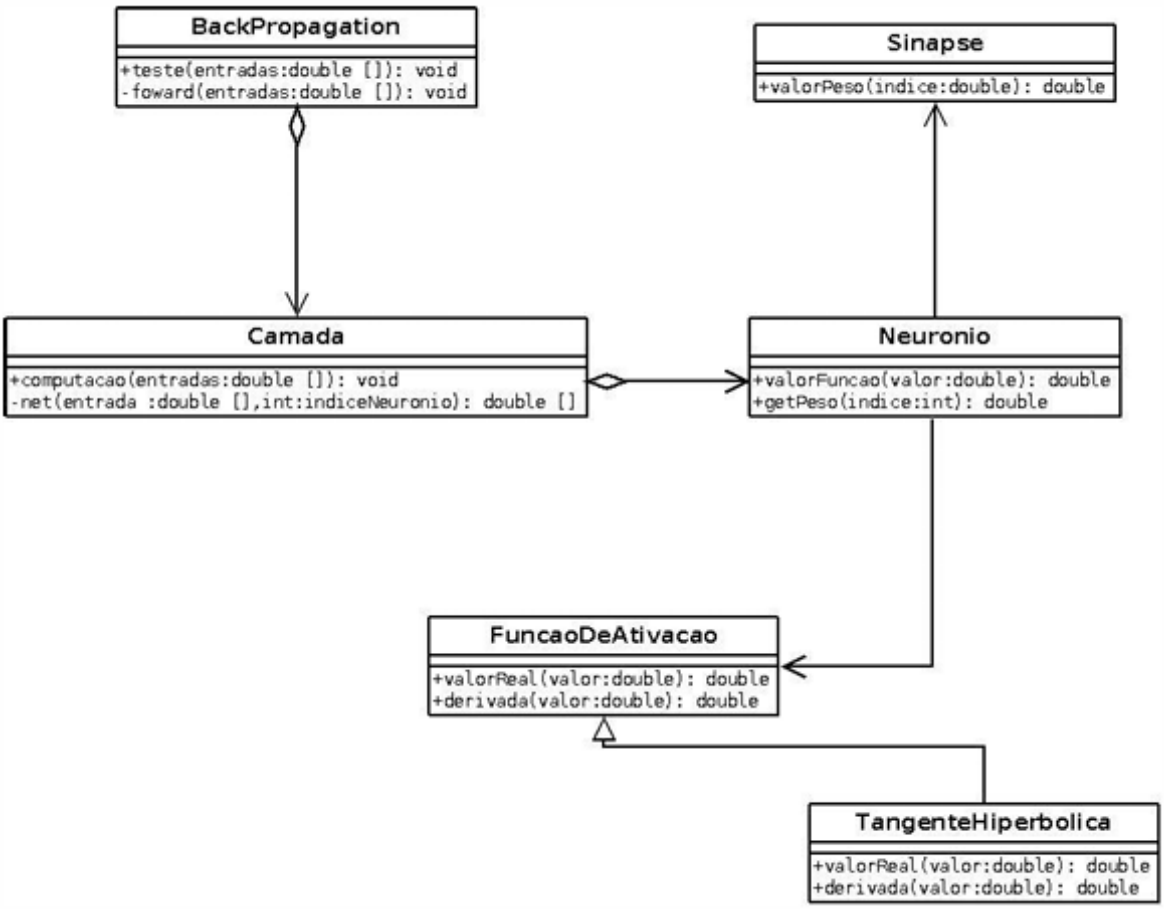


Figure 9 – Flowchart representation of the ANN used.

As for the training time several factors may influence the duration, but we always use some stopping criterion. In the backpropagation algorithm the stopping criterion is not well defined and it is usually used a maximum number of cycles. But it must be considered the average error rate per cycle and the generalizability of the ANN. It may happen that at a given moment the generalization of the training process start to degenerate, in other words, causing a loss in the ability of generalization. So we must find a good stopping point with minimum error and maximum capacity of generalization. The flowchart of the backpropagation ANN for control and automation of the multienzyme biosensor device for monitoring the blood metabolites glucose, cholesterol and urea.

4.5.4 Tests

During this phase the test set is used to determine the performance of the ANN with data that were not previously used. The performance of the ANN at this stage is a good indication of its actual performance. Other tests should be considered such as analyzing the behavior of the ANN using special entries and analysis of current weights of the ANN, because if there are very small values the associated connections can be considered insignificant and thus be eliminated. Conversely, values that are greater than others could indicate that there was a loss in the ability to generalize the ANN.

4.5.5 Integration

Finally with the ANN trained and evaluated, it can be integrated into an operating environment system of

the application. For increased efficiency of the solution this system should include usability facilities as a convenient interface and an ease acquisition of data through spreadsheets, interfaces with signal processing units, or standard files.
We should periodically monitor the system performance also network maintenance should be done when needed or indicate the need of re-training to designers. Further improvements can be suggested when users are becoming more familiar with the system, these suggestions may be very useful in new versions or new products.

5. ANN DEVELOPMENT FOR BLOOD METABOLITES ANALYSIS

The biosensors are a promising tool to supplement the clinical analysis techniques due to its properties such as selectivity, low cost of construction and storage, potential for miniaturization, making it easy to build simple and portable equipment where the patient himself could do all the monitoring.
The reading of recent literature shows that more and more problems in the area of analytical chemistry, such as the concentration analysis of components of a mixture, has been solved using ANNs, and it was observed that the multilayer perceptron using backpropagation algorithm has been the most widely used. This way, for teaching purposes an ANN was built, it uses the supervised learning and a feedback algorithm that is weights are modified during the computation so that the next weight will be given in terms of the current, the difference between the desired output and current output and the so-called learning rate. The development and testing process of the ANN will be described below.

The experimental procedure was basically divided into three stages. The first stage consisted by obtaining the reaction curves of the blood metabolites with the chemical components of the biosensor. In the second stage the data representing the curves were normalized and presented to the ANN. The third and final stage consisted of comparing the response data to validate the developed ANN and the method of analyzing curves proposed in this paper.

5.1 Obtaining Curves

The curves shown in this paper are obtained from the chemical reaction between the conductor and the blood metabolites present in the biosensor. As in any chemical reaction there are electron transferences, we used a Keitlhey voltage/current source to obtain each concentration curve of each of the three blood metabolites, a model 237 (K237) was used as shown in Figure 10, this voltage source is capable of responding which electric current was produced in a chemical reaction in a given time. This response is given as a time curve in the y-axis, measured in seconds, and electric current in the x-axis, measured in amperes.

The Keitlhey K237 is a source unit that measure with a high precision, essential to the realization of voltage measures between 10V to 1100 V, and current measures from 10 fA to 100 mA where these ranges are fundamental measures of low signals and any others that requires precision. For automated control of any data acquisition, this instrument has IEEE-488 standard, which allows its programming via computer. With all these features this measuring unit source accurately characterize the environment in which a patient will measure the concentrations of their blood metabolites using a common amperometric meter as previously shown.

Figure 10 – Keitlhey measuring unit source model 237.

For each hemometabolite - cholesterol, urea and glucose - 181 samples were made starting at 0.2 mM (0.2 mmol / L) then 0.4 mM (0.4 mmol / L) and increasing the concentration with 0.2 mM for each sample up to a concentration of 36.2 mM (36.2 mmol / L). This way for all three blood metabolites we will be covering the concentrations for medical interest, because the normal concentration of these blood components are shown in Table 1. Is worth remembering that:

Urea > 214 mg / dl (35.6 mmol / L) is indicative of acute renal failure.

Glucose < 45 mg / dl (2.52 mmol / L) neurological symptoms of hypoglycemia, which can extend from a decrease in cognitive function even lead to unconsciousness. Glucose > 450 mg / dl (25.2 mmol / L) diabetic coma due to lack of insulin. Cholesterol > 240 mg / dl (6.2 mmol / L) serious risk of heart disease.

Table 1 – Concentration of blood metabolites for medical interest.

Hemometabolite	Concentration of problems in humans (adults of both sexes)
Glucose	When the fasting glucose exceeds 11.2 mmol / L (200mg/dl of blood)
Cholesterol	<200 mg / dl (5.2 mmol / L) normal total blood cholesterol 200-239 mg / dl (5.2 to 6.2 mmol / L) limit of total cholesterol > 240 mg / dl (6.2 mmol / L) higher total cholesterol
Urea	When the urea concentration exceeds 8.3 mg / dl (1.4 mmol / L)

Figure 11 – Biosensor used for the reaction with hemometabolite.

Each of these concentrations was placed to react with the biosensor, shown in figure 11, producing curves as shown in Figure 12, generated through the measuring source.

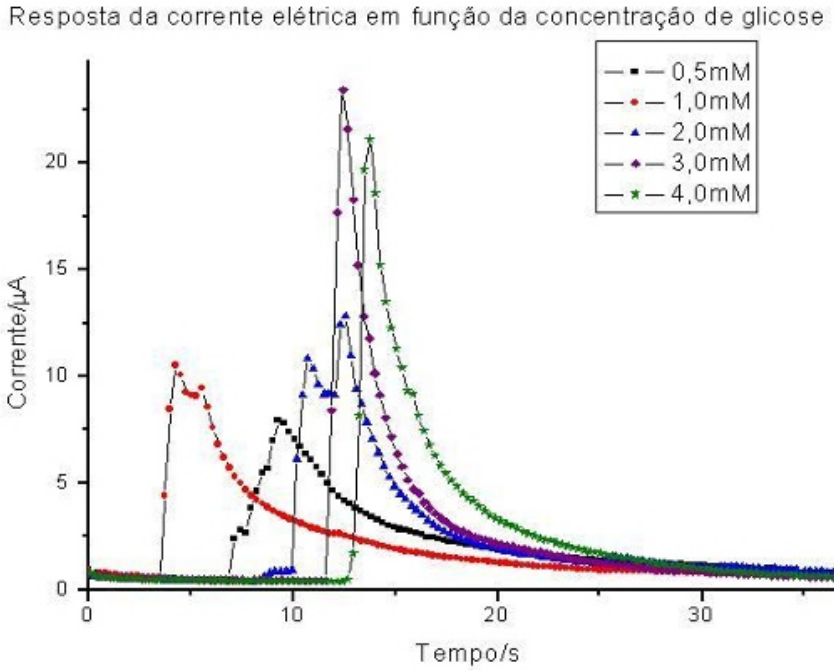


Figure 12 – Response of the electric current as a function of glucose concentration.

5. 2 Data preparation

With the curves of current vs. time, as shown in Figure 13, the data representing the curves were standardized to be presented to the ANN to be tested and validated. To standardize the data curves the following techniques were used:

1. Standardization by the range of variation, where the point to be standardized is subtracted by the lowest value point on the set and divided by the difference between the highest and the lowest points of the set.

$$y = (x - \min) \div (\max - min) \qquad (4.1)$$

2. Standardization by standard deviation, where the point to be standardized is subtracted by the average of the set of points and divided by the standard deviation of the set of points.

$$y = (x - \mu) \div \sigma \qquad (4.2)$$

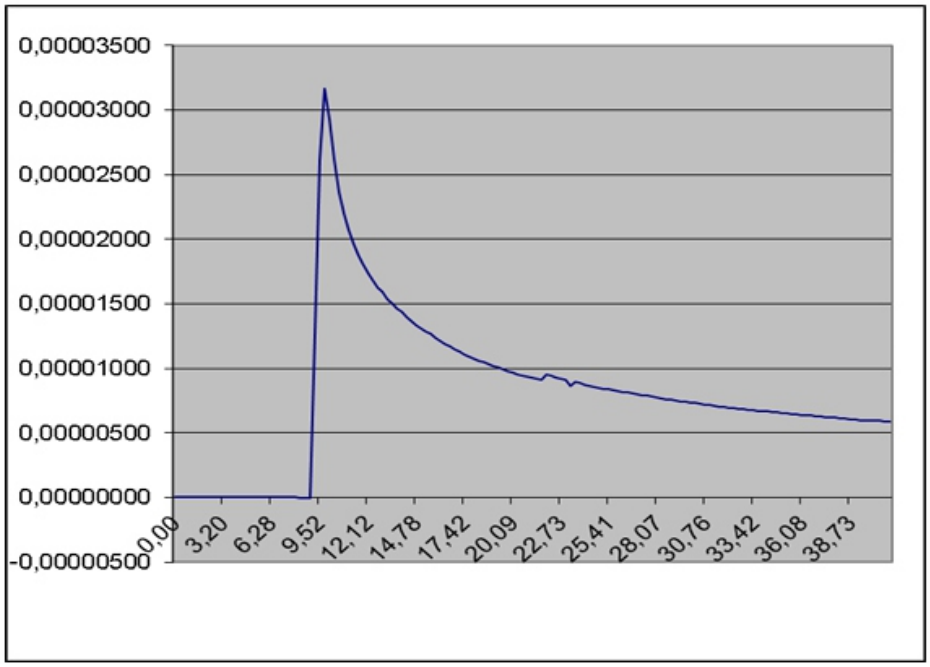


Figure 13 – Chart showing the current response time for the reaction of 20mM

Figure 13 shows the values for the reaction around 20 mM cholesterol with the reagent of the biosensor. In the y-axis is time in seconds of the chemical reaction and the x-axis is the electric current produced measured in uA. The graphs presented in Figures 14 and 15 have the same data described in Figure 13, but their axes are respectively formatted using the standardization by the range of variation and the standardization by standard deviation.

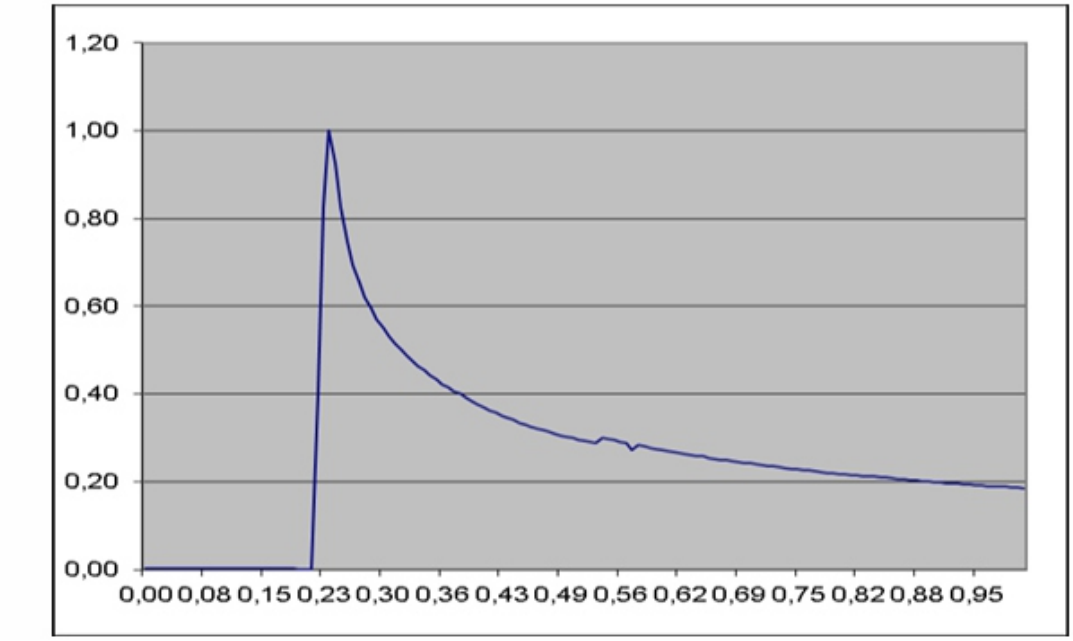


Figure 14 – Chart showing the current response time for the reaction of 20mM with axes standardized via standardization by the range of variation

Figure 11 – Biosensor used for the reaction with hemometabolite.

Each of these concentrations was placed to react with the biosensor, shown in figure 11, producing curves as shown in Figure 12, generated through the measuring source.

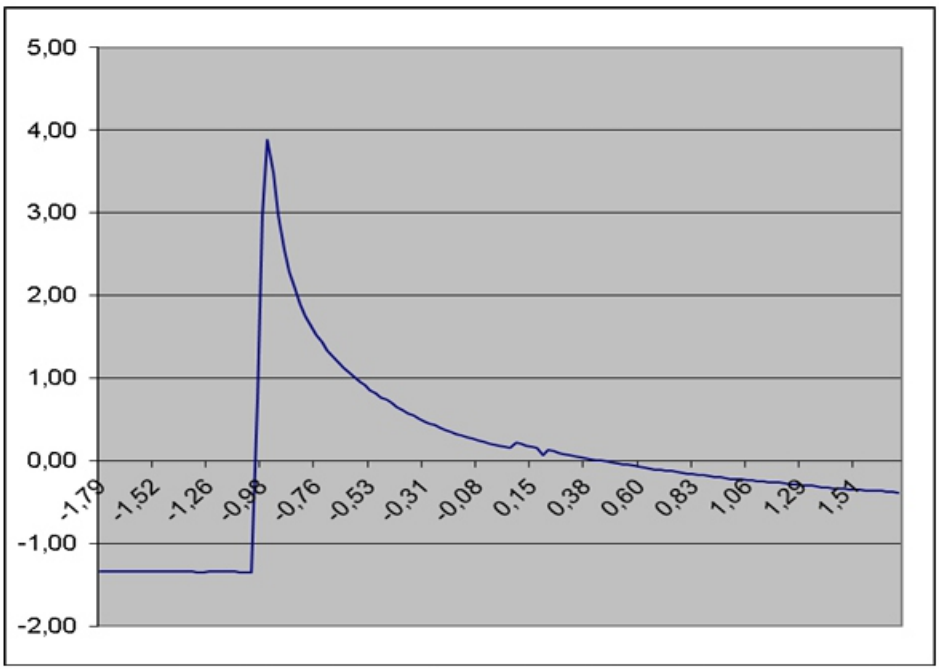


Figure 15 – Chart showing the current response time for the reaction of 20mM with axes standardized via standardization by the standard deviation.

The concentration values of the curves were also standardized using the same techniques. The blood metabolites were also identified numerically, but as in this case the identification job of the ANN is to classify the curve analyzed in one of the three classes of blood metabolites that are being studied in this article, the desired response for the classification is shown in Table 2.

Table 2 – Desired response for the classification of curves in three classes of blood metabolites.

	NEURON 1	NEURON 2	NEURON 3
CHOLESTEROL	1	0	0
UREA	0	1	0
GLUCOSE	0	0	1

5.3 Determination of the ANN structure

After the data is ready for presentation to the ANN, it was necessary to determine what structure of the ANN we were going use, how many nodes in the input layer, or how many input variables is needed for the ANN accurately determine what concentration of blood metabolites and which hemometabolite is being measured, and also in the hidden layer of the network how many nodes are needed. In addition to validate the ANN software that was developed during this work for its future use in other studies. For this purpose we used the Nets 3.0 and MATLAB R2008a, two highly renowned programs that works with neural networks.

5.3.1 Data separation for the formation of testing, validation and control groups

As a beginning of testing, due to the large number of curves that were available, 181 (one hundred and eighty-one) curves for each hemometabolite, the curves were divided into three groups. A group called test, with 327 (three hundred twenty-seven) curves with 109 (one hundred and nine) curves for each hemometabolite that was used for the learning of the ANN, 60% (sixty percent) of all curves. Another group called validation with 108 (one hundred and eight curves), 20% (twenty percent) of all curves, 36 (thirty six) curves for each hemometabolite used for testing (to verify that the values of free parameters of the ANN, synaptic weights, momentum and learning rate causes the ANN to determine which hemometabolite is being measured and what is its concentration) the ANN. Also another group called control, also with 108 (one hundred and eight) curves with 36 (thirty six) curves for each hemometabolite, 20% (twenty percent) of all curves, to ratify the free parameter values obtained, the table 3 summarizes the data curves.

Table 3 – Formation of validation, control and group tests.

	CHOLESTEROL	UREA	GLUCOSE	TOTAL
TEST	109	109	109	327
VALIDATION	36	36	36	108
CONTROL	36	36	36	108
TOTAL	181	181	181	543

5.3.2 Presentation of data to the ANN for tests

For testing purposes the data were presented as follows:

The training mode is sequential, this means that after each presentation of a set of example the error backpropagation occurred for the correction of the synaptic weights.

During the training phase of the ANN cholesterol, urea and glucose curves were alternately presented to the ANN, but respecting an order of increasing concentration value.

For validation and control the curves were presented randomly to the ANN.

Data using both standardization were presented, but in the first tests the standardization by the range of variation method was proved to be better be reaching better results, once this standardization produce results with values between 0 (zero) and 1 (um) which facilitates the ANN process of learning (SCHALKOFF, 97), furthermore, the data was standardized by the standard deviation were not used in the test.

5.3.3 First test

For all tests, the ANN used is the multilayer perceptron, feedforward, using the backpropagation algorithm to adjust the synaptic weights. For the first test we used the following structure of the ANN:

Input layer with two neurons to receive the ordered pairs (electric current, time).

A hidden layer with 10 (ten) computational neurons and with a nonlinear sigmoid activation function.

The output layer with 4 (four) computational neurons and nonlinear sigmoid activation function.

Learning rate equals to 0.6 and momentum equals to 0.3.

We used three different neural networks, one of them developed using MATLAB R2008a a second one using the Nets 3.0, and a third one using the ANN software developed during this work. The ANNs developed using MATLAB R2008a and Nets 3.0 are being used in this work also to check the performance of the ANN software developed in this work, this software is being called SJRNA in the text. The results obtained in the first trial are listed in Table 4.

Table 4 – Results of the first test

Simulator	Seasons	Hits of concentration	Hits of Hemometabolite
JRNA	3500	0%	100%
Nets 3.0	3500	0%	100%
MATLAB	3500	0%	100%

The 100% value of accuracy is considered as follows, for example:

1.Considering the concentration of 5.2 mmol and 5.4 mmol its standardized values are 0.138889 and 0.144444.

2. It is considered that there was an error by the ANN when the difference of the produced value by it is greater than 0.001, this error is enough so that the ANN can accurately determine with precision the concentration value, once the smallest difference between the two concentrations in sequence is 0.004, and to determine which class of hemometabolite the curve belongs this error is quite satisfactory, because there are only three classes to be checked and the worst output produced by the network had the following values: (0.9989 / 0.00108 / 0.001783) produced by the first test.

The three ANNs used in the first test were proven to be very effective concerning the identification of which hemometabolite belongs to the points of the analyzed curve, but were unable to identify the concentration of each curve, this was due to the existence of a large amount of ordered pairs (current, time) that belonged to more than one same curve. And another problem was the delay in making a training because each curve to be presented consists of 150 ordered pairs (current, time), this way the

need to reduce the amount of data to be presented to the ANN was observed and also the need to introduce some other variables in order to the ANN have a reasonable performance in identifying the concentration.

5.3.4 Second Test

For the second test, due to the need to decrease the amount of ordered pairs representing each curve and the introduction of new variables to improve the performance of the ANN the following changes in the training set and structure of ANN were made:

1. For each curve was calculated the area of the curve peak, as shown in Figure 16, and the breadth and these two data began to be used as input parameters of the ANN by increasing the number of entries from two to four.
2. To decrease the number of ordered pairs representing each curve only the ordered pairs from the peak were used, this way each curve is going to be represented by 40 (forty) ordered pairs (current, time).

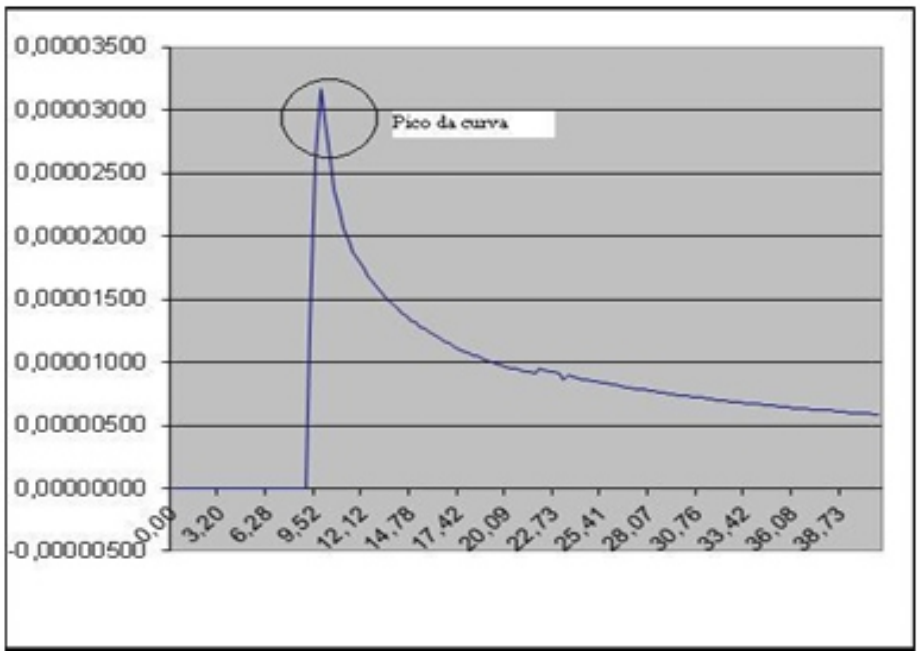


Figure 16 – The curve peak is shown in evidence.

With the changes the ANN configuration was as follows:

Input layer with 4 (four) neurons to receive the parameters of the peak area, curve peak width and the ordered pairs current and time.

Hidden layer with 10 (ten) computational neurons and sigmoid nonlinear activation function.

The output layer with 4 (four) computational neurons and sigmoid nonlinear activation function.

Learning rate equals to 0.6 and momentum equals to 0.3.

The results with the introduction of these changes can be seen in Table 5 below.

Table 5 – Results of the second test.

Simulator	Seasons	Hits of Concentration	Hits of blood metabolites
SJRNA	1300	97%	100%
Nets 3.0	1500	99%	100%
MATLAB	1100	99%	100%

The 100% value of accuracy is considered as follows, for example:

Considering the concentration of 5.2 mmol and 5.4 mmol its standardized values are 0.138889 and 0.144444.

It is considered that there was an error by the ANN when the difference of the produced value by it is greater than 0.001, this error is enough so that the ANN can accurately determine with precision the concentration value, once the smallest difference between the two concentrations in sequence is 0.004, and to determine which class of hemometabolite the curve belongs this error is quite satisfactory, because there are only three classes to be checked and the worst output produced by the network had the following values: (0.9989 / 0.00108 / 0.001783) produced by the first test.

The new data show that the changes that were made are essential on determining the values of the curves, once the worst result the ANN had was not capable of distinguish only 16 curves that had concentration with the difference of only 0.2 mmol.

5.3.5 Third Test

In the third test all changes made in the second test were retained, but with the difference that we used two different ANNs, one to determine the concentration and another one to determine which class of hemometabolite belonged the analyzed curve. The ANN used to determine which class belonged the curve to be analyzed was given as follows:

Input layer with two neurons to receive the ordered pairs (current, time).

A hidden layer with 10 (ten) computational neurons and sigmoid nonlinear activation function.

The output layer with 3 (three) computational neurons and sigmoid nonlinear activation function.

Learning rate equals to 0.6 and momentum equals to 0.3.

The results are shown in the Table 6 below.

Table 6 – ANN test results to determine the class of the curve

Simulator	Seasons	Hits of Hemometabolite
SJRNA	1500	100%
Nets 3.0	1300	100%
MATLAB	1100	100%

The ANN used to determine the concentration of the analyzed curve was given as follows:

Input layer with 4 (four) neurons to receive the parameters of the peak area, curve peak width and the ordered pairs current and time.

Hidden layer with 10 (ten) computational neurons and sigmoid nonlinear activation function.

The output layer with 1 (one) computational neuron and sigmoid nonlinear activation function.

Learning rate equals to 0.6 and momentum equals to 0.3.

The results obtained with the introduction of these changes can be seen in the Table 7 below:

Table 7 – ANN test results to determine the concentration curve

Simulator	Seasons	Hits of Concentration
SJRNA	1800	100%
Nets 3.0	1500	100%
MATLAB	1300	100%

The separation into two ANNs was efficient because they began to accurately hit all the concentrations and blood metabolites classes.

6. CONCLUSION.

By analysis of the tests can be concluded that:

The process described in this article to classify and determine the concentration of hemometabolites proved to be efficient, mainly when using specialized ANN to classification and determination of concentrations, as show the third test.

Using the same methods for obtaining curves of chemical reactions discussed in this article. To determining concentration of hemometabolite, the ANN multilayer perceptron type that uses backpropagation algorithm to adjust the synaptic weights proved to be efficient and can used to determine concentrations of reagents of others chemical reactions.

REFERENCES

[1] BURTIS, C. A.; ASHWOOD, E. R. Tietz: Fundamentos de Química Clínica. 4 ed. Rio de Janeiro: Guanabara, 1998, 836 p.

[2] CLARK, L.; LYONS, C. Annals of the New York Academy of sciences, n. 29, p. 102-123, 1968.

[3] WILSON, G.S.; GIFFORD, R. Biosensor for real-time in vivo measurements. Biosensors and Bioeletronics, n. 20, p. 23882403, 2005.

[4] FERNANDES, E.G.R. Biossensores nanoestruturados para monitoração de glicose. 2005. 175 f. Dissertation (Materials Engineering Master Degree) – Federal University of Itajubá. Advisor: Prof. Dr. Álvaro A.A. de Queiroz.

[5] STRYER, L.; BERG, M.J.; TYMOCZKO, L.J. Bioquímica. Guanabara Koogan, 2002.

[6] Cinética enzimática. Disponible at <http://pt.wikipedia.org/wiki /Cin%C3%A9tica_enzim %C3%A1tica> January in 25. 2012.

[7] BOYER, R. Concepts in Biochemistry, 2 ed., New York, Chichester, Weinheim, Brisbane, Singapore, Toronto: John Wiley & Sons, 2002, p. 137–138.

[8] FERREIRA, J. P. M. Segredos da Catalise enzimática. Boletim de Biotecnologia.

[9] Universidade Católica Portuguesa, Porto. p. 22-29.

[10] DOMINGOS, J.B.; LONGHINOTTI, E.; MACHADO, V.G.; NOME, F. A química dos ésteres de fosfato. Química Nova, São Paulo, v. 26, n. 5, Set./Out. 2003.

[11] CARRIJO, R.M.C.; ROMERO, J.R. Oxidações catalíticas e eletrocatalíticas de substratos orgânicos. O cério como oxidante. Química Nova, v. 23, n. 3, 2000.

[12] SCHMIDT, H.L.; GUTBERLET, F.; SCHUHMANN, W.; Sens. Actuators B. p. 13-14, 366 p., 1993.

[13] OKAWA, Y.; NAGANO, M.; HIROTA, S.; KOBAYASHI, H.; OHNO, T.; WATANABE, M. Bios. Bioelect. v. 14, 229 p., 1999.

[14] SOTOMAYOR, M.D.P.T.; KUBOTA, L.T. Enzymeless biosensors: uma nova área para o desenvolvimento de sensores amperométricos. Química Nova, v. 25, n 1, p. 123-128, 2002.

[15] BERCHMANS, S.; VIJAYAVALLI, R. Surface Modification of Glassy Carbon by Riboflavin.

Langmuir. n.11, p. 286-290, 1995.
[16] COIERA, E.W. Inteligência Artificial na Medicina Disponible at <http://dfm.ffclrp. usp.br/infobio/ceib /texto4.php> Accessed in January in 25. 2012
[17] GONGORA, A.D. O Que é Inteligência Artificial. Disponible at <http://www.pr.gov.br/batebyte/edicoes/ 2002/bb119/estagiario.htm> Accessed in January in 25. 2012.
[18] Uma introdução às Redes Neurais. Disponible at http://www.din.uem.br /ia/neurais/ Accessed in January in 25. 2012.
[19] Neural Networks. Disponible at <http://www.zsolutions.com /Neural.htm> Accessed in January in 25. 2012.

ISSN 2210-142X

M-based Filter Design for Communication and Imaging Systems

Akitoshi Matsuda1 and Shinichi Baba2
1Dept. of Automotive Science, Kyushu University, Fukuoka 814-0001, Japan
matsuda_aki@slrc.kyushu-u.ac.jp
2Kyushu Embedded Forum, Fukuoka 814-0001, Japan
shinichi.baba@nifty.com

ABSTRACT

The MATLAB M language provides constructs for the simple representation of functions, and has been used extensively for algorithm development and verification in communication and imaging systems design. The M language processes entire arrays simultaneously, and it is expected that design based on M can be implemented faster and with the same or higher quality as compared with design based on other languages. This paper presents a design methodology using M-based design flow for signal processing filters. The results of this design methodology demonstrate the high efficiency of the M-based design flow as compared with conventional functional modeling and manual coding using a hardware description language.

Keywords: M language; algorithm development; verification; filter design; signal processing.

I. INTRODUCTION

In the design of current large-scale and highly functional embedded systems, enhancement of design quality often entails an increase in the time and cost of developing the hardware of embedded systems. Design methods based on MATLAB and Simulink focus on improving the efficiency of developing increasingly complex hardware as well as algorithm design in specification-level design [1]. Here, a design method is proposed based on the MATLAB M language. In this paper, model-based design (Fig. 1) represents hardware models (multipliers, adders, multiplexer, etc.) as combinations of hardware circuitry. The adoption of this representation facilitates the understanding of hardware configurations and architectures in comparison with representation based on the C language or a hardware description language (HDL). Fig. 2 shows coefficient matrices used in a Sobel filter, which is commonly used for edge detection in image processing. In the example, two operators (one for horizontal and one for vertical line detection) are represented in the M language [2].

In general, these matrix operations provide a description similarly to mathematical formulas in M-based design, which also facilitates the understanding of the processes in algorithm development. In addition, these two design methods support automatic simulation and generation of HDL code. Thus, their adoption in hardware development is expected to reduce the need for manual coding as compared with traditional methods in design, implementation, and verification at the algorithmic level. A “visualization” feature can also be added to the entire development process, which is also expected to reduce turn

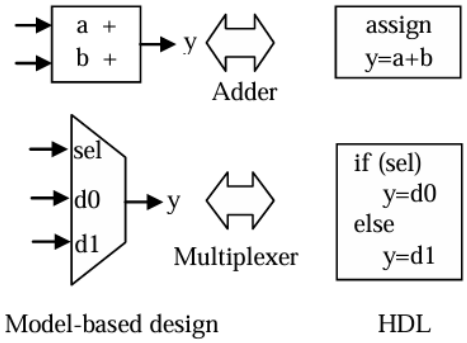


Figure 1. Hardware model.

```
% sobel coef
    sobel_x = [ -1  0  1 ;
                -2  0  2 ;
                -1  0  1 ];

    sobel_y = [ -1  -2  -1 ;
                 0  0  0 ;
                 1  2  1 ];
```

Figure 2. Example of matrix representation using M. around time (TAT) and to improve the overall design quality.

Our paper is structured as follows: Section 2 describes the background to M-based design. Section 3 gives an overview of design methodology in both M-based design and C-based design. Section 4 presents an outline of the image processing algorithm used in this study. Section 5 describes a set of experiments that we conducted on diverse real-world applications on the filter, and furthermore presents the design methodology for the implementation and simulation of Mbased design. After that, we discuss the results in Section 6 and state our conclusions in Section 7. Lastly, Section 8 discusses future work.

II. BACKGROUND

In this paper, we introduce M-based design methods implemented in MATLAB and Simulink. Model-based design methods implemented in MATLAB and Simulink will be described elsewhere. Increasing the productivity of hardware implementation from the stage of algorithm design and exploration should help to improve TAT, as well as the overall quality of the entire embedded system. From the results of several trials, the M language has been recommended for algorithm design in earlier stages of hardware development, which includes mathematical operations for signal processing (linear operations, complex arithmetic operations, etc.) [3].

Behavioral synthesis with the M language is expected to be advantageous for hardware design since M is highly efficient for developing hardware design techniques that apply synthesis at a high level of abstraction and language input. This efficient M-based design method was applied in this work with the aim to implement signal processing algorithms in hardware. While an M-based design method was utilized in design development, the HDL code for hardware implementation was generated with a language conversion tool in order to automate the design flow. An efficient hardware design method was used that facilitates the trade-off analysis of hardware design in order to visualize the overall structure and to remove unnecessary variables. In this paper, we report a case study on the application this design

method to hardware design flow.

III. DESIGN BASED ON M AND C BASED LANGUAGES

Various methods have been proposed for reducing the time necessary for designing embedded system hardware. Schliebusch et al.'s technique [4] has been proposed as the most versatile in the case of design methods based on C-based languages (such as C/C++, SystemC, and SpecC) [5,6,7]. Techniques for automatic conversion from C-based languages into HDL by means of behavioral synthesis tools have been developed [8]. However, since C-based languages are originally software-oriented, the utilization of these behavioral synthesis tools for this task poses considerable difficulties.

Furthermore, in system-level design, C-based languages do not implement the data structure and operators necessary for the target operation, which further increases the cost and effort associated with design implementation since these operations must be created separately in accordance with the required functionality. For example, in image processing of files in various formats, a program must be written to create a stream of image data. For this purpose, input/output functions are embedded into the M language. Specifically, image files are typically stored in various formats such as BMP, JPEG, and TIFF, and pixel data can be read as an array by using the built in functions in MATLAB [9]. The following items represent several major advantages of the M language over C-based languages in regard to hardware design:

Simplified test bench generation.

Straightforward creation of fixed-point models.

Quick response to design constraints.

In other words, the M language provides a rich simulation environment for functional verification. Therefore, if the advantages of the M language are utilized in the design process, the flow from algorithm development to hardware design can be simulated rapidly and accurately. Moreover, the automatic conversion of M source code into valid HDL code is expected to enable the design of high-performance and high quality hardware in a seamless process encompassing all steps from algorithm development to hardware implementation, while maintaining a high level of abstraction.

IV. ALGORITHM DEVELOPMENT

In this section, the image processing algorithm used in this study is described. In image processing, there are a variety of purposes that require detecting a boundary (edge) of an area in the image. An edge can be detected by using the differential operator for the change in pixel brightness, because this change is large at the region boundary. Another possibility is using a difference operation instead of a differential operation in processing the digitized image. Here, a coefficient combination for performing the differential operation is referred to as a differential filter. Differential filters can be classified as firstderivative filters (Sobel filter, Prewitt filter, etc.) and second derivative filters (Laplacian filter, etc.).

A linear filter is generally expressed in an input image f(x,y) and filter function h(x,y) with a convolution integral as follow:

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x-x, y-y)\, f(x,y)\, dxdy \qquad (1)$$

In an actual computer, a digital calculation is performed using a digital image. Therefore, equation (1) is expressed in discrete form as follows:

$$g(i,j) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h(i-k, j-l)\, f(k,l)\ . \tag{2}$$

Here, M and N are the number of pixels along the x- and y directions of each image. Moreover, a neighboring region in the filter is a rectangular region around the target image element.

$$g(i,j) = \sum_{k=i-m_1}^{i+m_2} \sum_{l=j-n_1}^{j+n_2} h(i-k, j-l)\, f(k,l)\ . \tag{3}$$

Furthermore, we define m, n, and h' as follows:

$$m = k - i;\ n = l - j;\ h'(i,j) = h(-i,-j)\ , \tag{4}$$

and

$$g(i,j) = \sum_{m=-m_1}^{m_2} \sum_{n=-n_1}^{n_2} h'(m,n)\, f(i+m, j+n)\ . \tag{5}$$

This section describes a differential filter. We calculate the differential in the image with respect to x and y:

$$\frac{\partial f}{\partial x} = f(i+1, j) - f(i, j) \tag{6}$$

and

$$\frac{\partial f}{\partial y} = f(i, j+1) - f(i, j)\ . \tag{7}$$

In the first-derivative filter, we then set up a differential whose magnitude is independent of direction:

$$\nabla f = \left\{ \left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right\}^{1/2}\ . \tag{8}$$

However, this calculation is nonlinear. The output filter is therefore a nonlinear filter [10]. In the second-derivative filter, the Laplacian of the second-order differential equation is shown below:

$$\nabla^2 f = \frac{\partial^2}{\partial x^2} f + \frac{\partial^2}{\partial y^2} f\ . \tag{9}$$

We present an overview of a design method based on the MATLAB M language. This particular section presents the development of an image processing algorithm for edge detection with the aid of the M language. In general, when investigating image features, it is necessary to perform feature extraction and filtering. Accordingly, edge detection, an important feature extraction process, is described here. This

algorithm uses Sobel edge detection of horizontal and vertical lines in images [11]. Typically, the Sobel filter uses coefficient matrices (shown in Table 1) to detect horizontal and vertical lines Fig. 3 shows the original image, while Fig. 4 shows the results of the edge detection process [12]. It is apparent that only vertical and horizontal lines are clearly detected in the respective steps of this algorithm. Combining the horizontal and vertical gradients (fx(x,y) and fy(x,y) respectively), the edge detector f '(x, y) can be calculated as follows:

$$f'(x,y) = (f_x(x,y)^2 + f_y(x,y)^2)^{1/2}. \qquad (10)$$

Equation (10) is calculated for each pixel, the resulting image shows sharp contours representing edges in the image. Thus, edge detection is possible by using a Sobel filter. In this calculation process, an example of an implementation of the Sobel filter algorithm in M is shown in Fig. 5, where the coefficient matrices used in the filter can be defined directly inside the function. The image data can be represented in the form of an array, such as (I) in the example, by using built-in file input functions.

Currently, the Verilog HDL and VHDL languages are used for detailed design of hardware, the C and C++ programming languages are used for software development, and system description languages or blocks are used for system design in order to integrate the specifications in a manner that satisfies the specifications. Thus, one factor that obstructs the efficient design of embedded systems is the use of different languages by the parties involved, namely, the hardware designer, the software designer, and the system designer.

In this regard, a model can be created automatically in order to determine the detailed specifications for hardware implementation from a high-abstraction algorithm developed by using the M-based design method, which can improve design productivity. In addition, simulation performance and modeling efficiency can also be improved when a reference model is created at a high abstraction level with the M language and a simulation is conducted with that model.

TABLE I. COEFFICIENT MATRICES USED IN SOBEL FILTERING

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Figure 3. Original image.

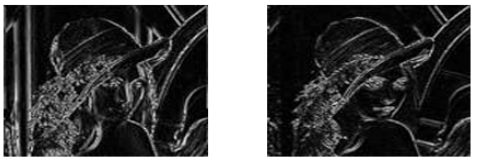


Figure 4. Vertical and horizontal feature extraction with Sobel filtering.

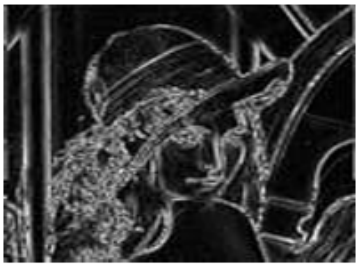


Figure 5. Combined results of vertical and horizontal feature extraction.

Therefore, if HDL code can be generated automatically from the M language, it would be possible to design a largescale integration (LSI) and field-programmable gate array (FPGA) in short TAT and without the need for manual HDL coding. In addition, the developed algorithms and the HDL code can be easily simulated, which improves the efficiency of functional verification. As a result, these measures would allow for the implementation of efficient verification and cosimulation environments for embedded systems.

V. EXPERIMENTAL RESULTS

A. Case study of image processing filter design

In this section, we report a case study where the M language was used in the design of the Sobel image processing algorithm described in the previous section. In this case study, an FPGA was used as the target device for designing a hard-

```
function [K] = edge_detector(I)
%% Parameter and coeff setting
% size
[nRows, nCols] = size(I);

% sobel coef
   sobel_x = [ -1  0  1 ;
               -2  0  2 ;
               -1  0  1 ];

   sobel_y = [ -1 -2  -1 ;
                0  0  0 ;
                1  2  1 ];

%% Edge Detection
% Initialize variables
K = zeros(nRows, nCols);
```

```
% Adding margin rows and columns
Img_temp = [zeros(nRows,1) double(I) zeros(nRows,1)];
Img_marg = [zeros(1,nCols+2); Img_temp;
                              zeros(1,nCols+2)];

% Main operation
for iRow = 1:nRows
   for iCol = 1:nCols
            ghs = 0;
      gvs = 0;

% Sobel filtering
    for  jRow = iRow:iRow+2
      for  jCol = iCol:iCol+2
       ghs = ghs + sobel_x(jRow-iRow+1,jCol-iCol+1)
                    * Img_marg(jRow,jCol);
       gvs = gvs + sobel_y(jRows-iRow+1,jCol-iCol+1)
                    * Img_marg(jRow,jCol);
       end
     end
    g = sqrt(ghs^2 + gvs^2);
   end

%Perform Thresholding
      if g >150
         K(iRow, iCol) = 255;
      else
         K(iRow, iCol) = 0;
      end
    end
 end
```

Figure 6. Example of algorithm development in M.

```
clear all;
clc;
close all;

% Stimulus
            I = imread('original.jpg');

% Edge Detector (HW part)
            K = edge_detector ( ...
              I(:,:,3) ...
               );
```

```
% Display result
        O(:,:,1) = uint8(K);
        O(:,:,2) = uint8(K);
        O(:,:,3) = uint8(K);

figure('Position',[100, 100, 1200, 500]);

        subplot(1,2,1)
        image(I)
        subplot(1,2,2)
        image(O)
```

Figure 7. Example of script in M.

TABLE II. PERFORMANCE RESULTS

Latency	Adder	Logic level	Frequency	Area
8	15	9	101MHz	4455 Gate

ware implementation of this algorithm. First, the description in M shown in Fig. 6 and 7, was simulated in MATLAB in order to validate the algorithm, where the verification was followed by the hardware development phase if the results agreed with the specifications. The constraints of the hardware implementation, such as fixed-point operations, the number of input and output bits and the performance frequency, were given, and behavioral synthesis was conducted. HDL was automatically generated from the M source code at the algorithmic level. Table 2 shows the circuit performance for reference. The most important constraint in this study was the operating frequency of the FPGA, which was set to 100 Mhz. The circuit was considered to be generated as desired when this constraint was met. Next, the design time was evaluated. The M language was used to examine the Sobel filter design specifications (including algorithm development). Thereafter, the time required for automatic generation of HDL code in M-based design was compared with the time necessary for manual HDL coding. In the M-based design method, which was based on the algorithmic level of abstraction, a simulation considering the hardware design was conducted. Thus, the process of converting floating-point operations into fixed-point operations

TABLE III. COEFFICIENT MATRICES USED IN PREWITT FILTER

-1	0	1
-1	0	1
-1	0	1

1	1	1
0	0	0
-1	-1	-1

TABLE IV. COMPARISON BETWEEN SOBEL AND PREWITT FILTERS

Item	Sobel	Prewitt
Max freq. (MHz)	100.9	102.9
Area (KG)	7.74	7.49
LUT (pieces)	393	374
Levels of logic	9	7
Run time (s)	438	441

(i.e., from M to HDL code) was automatic. In addition, the test benches of the HDL description were also generated automatically and simultaneously with the generation of HDL code, and the time required for simulation verification was reduced. Note that hardware and software design can be validated in parallel with the design requirements after the specifications are in place. Doing so can reduce the number of design iterations and re-spins and thus reduce the overall development time. Moreover, the reusability of the operating environment increased by continuing the M-based design process, which presents further possibilities for reducing the development time [13]. Next, a Prewitt filter was designed by using M-based design. This design was easily accomplished by simply changing the values of the matrix elements of a Sobel filter.

Specifically, some values of the matrix elements in Table 1 are changed to those in Table 3. Table 4 compares these two filters. The performance (maximum operating frequency) and area (Kgate number) of these two filters were nearly the same. The run time in the table indicates the implementation time from register-transfer level generation using the M language. In short, once a Sobel filter is designed using an M-based language, a Prewitt filter can be designed within approximately 500 s. This remarkable feature is also effective for design reuse in M-based design flow.

B. Case study of comunication filter design

We also conducted a case study involving the design of a digital filter used extensively in various embedded systems. As in the previous section, the target device of the design case study was an FPGA. In this case, the filter algorithm was developed in M, after which it was verified at the level of floating-point operations. When the behavior of the algorithm

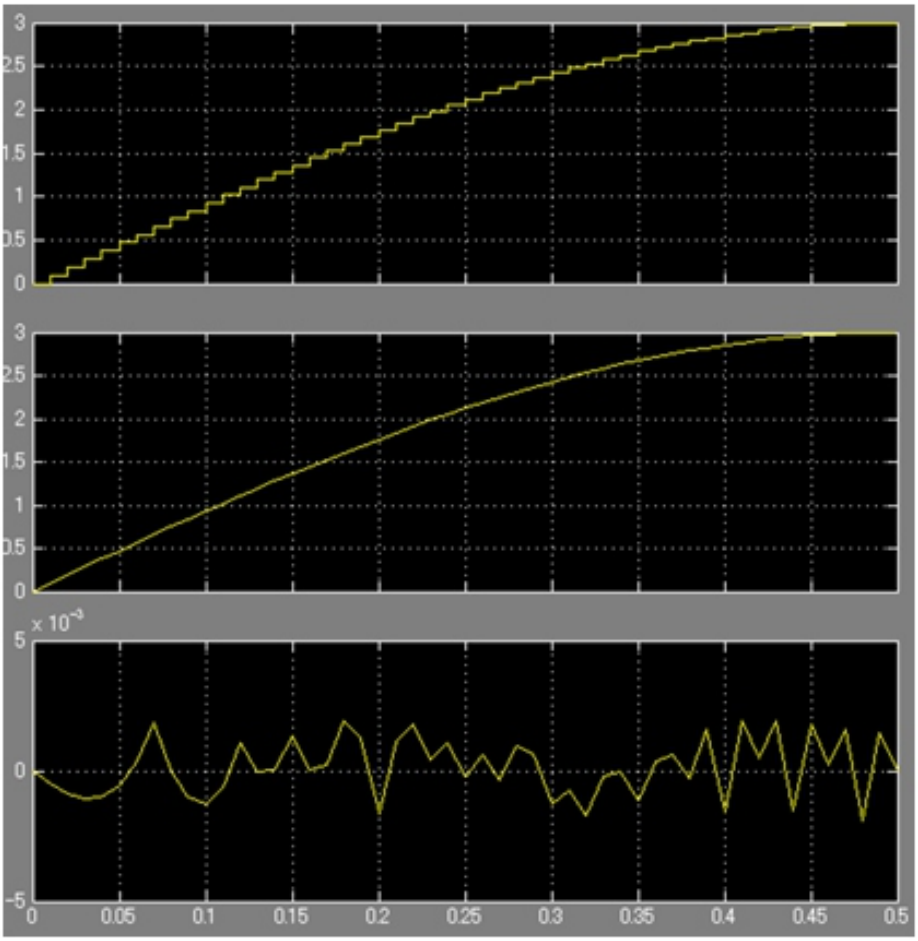


Figure 8. Comparison of fixed-point results (top) and floating-point results (middle), and the resulting quantization error (bottom).

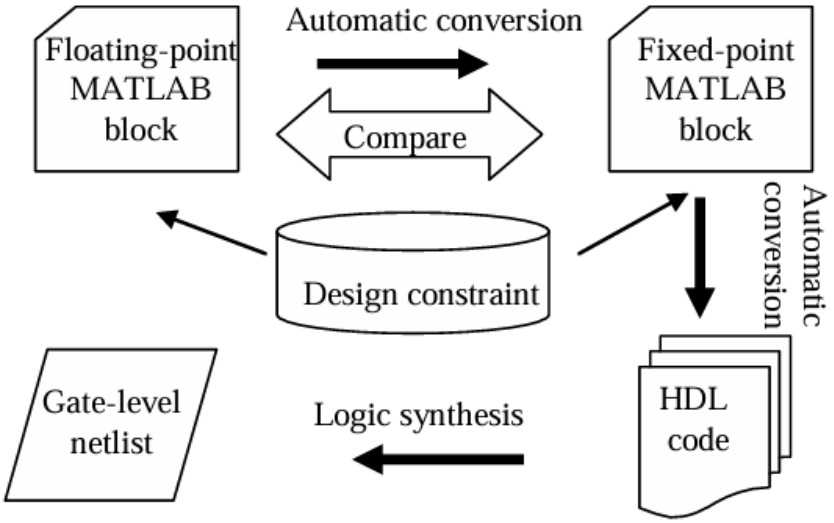
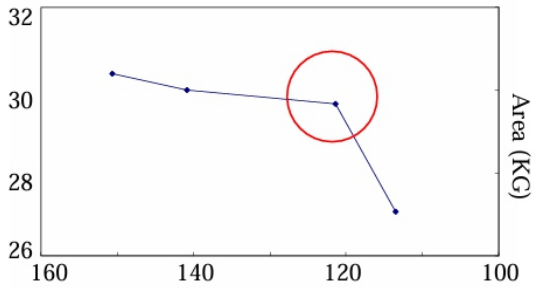


Figure 9. Digital filter design flow.

•



Performance frequency (Mhz)

Figure 10. Trade-off analysis results.

was satisfactory, an automatic conversion into fixed-point operation was performed in order to obtain the respective hardware design, where the fixed-point operations were described as fixed integer and decimal parts of a bit sequence. Quantization error arises when approximating real numbers with a variable decimal point, for example, when comparing the results of floating-point and fixed-point calculations. The magnitude of this error must be examined in hardware design, and MATLAB can be used for rapid and straightforward computation of this parameter. Fig. 8 shows the results for the quantization error associated with fixed-point and floating point operations, where the fixed-point simulation results are shown in the top panel, the floating-point simulation results are shown in the middle panel, and the quantization error is shown in the bottom panel. Clearly, the quantization error is within the interval of ±3/1000. Additionally, MATLAB can display and compare three simulation results simultaneously on the same screen, which is convenient when verifying the results.

Here, we describe a case study of designing a Kalman filter, which is often used in car navigation systems as well as in video processing. First, the Kalman filter algorithm was developed in M. Fig. 9 shows the hardware design flow, where it is clear that once developed, the floating-point model in M at the algorithm development phase automatically proceeds to the hardware implementation phase in the form of a gate-level netlist.

In general, the two major constraints in this case are speed (frequency) and cost (device area) for digital filter design, in other words, the desired results are high speed and low cost. However, those two constraints are in a trade-off relationship, and they must be investigated by changing a number of parameters and performing several design iterations with each set of parameters. This design flow can be automated in order to promptly obtain trade-off analysis results in a simple manner by examining various combinations of constraints. Fig. 10 shows the analysis results for this case study, where the design constraints were a frequency over 100 MHz and an area smaller than 30 KG, and it is clear that the optimal result is attained at a frequency of around 120 MHz Thus, the M-based design method can be used for performing faster and easier trade-off analysis without any changes to the M-based description at the algorithmic level.

C. Case study of FIR filter design and verification

Next, we introduce a case study where M-based design is used in the verification of a finite impulse response (FIR) filter, which is one of the most commonly utilized digital filter types in mobile phones, receivers, and other devices. This time, the target of the design is a 16-dimensional FIR low-pass filter with given specifications. The main specifications are a sampling frequency of 44 kHz and a cut-off frequency of 2 kHz. Other filter factors, such as response type, frequency specification and amplitude, can be easily set in MATLAB.

In the M-based design environment, if a parameter is defined as a user variable, then it is considered that the parameter can be changed freely, and therefore the design

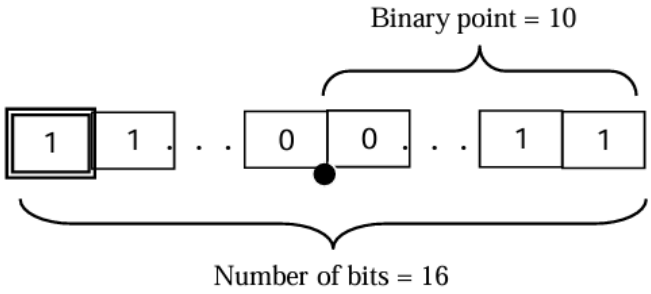


Figure 11. Setting the number of input bits.

TABLE V. TRADE-OFF ANALYSIS BASED ON THE NUMBER OF INPUT BITS

Number of bits	8	12	16	20
Freq. (MHz)	142	136	133	110
Area (Gates)	923	1170	1215	1890
Slices (pieces)	67	83	103	144

specifications can be modified. Here, the process involves designing the filter hardware and focusing on the number of input bits, which is one of the most important variables. As described earlier, the input consists of a number of bits and a binary point. If these parameters are provided as data with variable length and decimal point position, they can be changed. In this regard, Fig. 11 shows an outline of this configuration. At the filter algorithm development phase, input signals are processed by floating-point operations, while fixedpoint operations are needed in the case of hardware design.

Therefore, the number of input signal bits must be fixed in the latter. Although the signal accuracy becomes higher as the number of input bits increases, the size of the designed circuit increases, and vice versa. As described in the previous sections, trade-off analysis is performed with respect to the number of input bits. Table 3 shows how the circuit performance varies based on the number of input bits.

At the design specification phase, the number of input bits was set to 12 or 16, and the corresponding results are shown in Table 3. As clear from the table, the effect of changing the number of input bits on the operation frequency and the area was negligible, and since the result with 16 bits indicates higher accuracy, the number of bits was set to 16. In this case, the data length (number of bits) was 16 bits, and the position of the decimal point (binary point) was set to 10 bits. Based on the trade-off analysis performed by applying the M-based design method, it can be concluded that the analysis results can be obtained promptly, thus reducing the amount of time required for development, and the design specifications can be verified with high accuracy [14].

VI. DISCUSSION

We have demonstrated the possibility for automating all processes from algorithm verification to system description and detailed design of hardware in a consistent manner by using a design method based on the M language. As a result, high quality system-level design can be accomplished in a comparatively shorter period of time than with conventional tools. The realization of such functionality has been

approached from various directions in the past, and currently the most commonly used design techniques are based on C based languages (high-level synthesis techniques). This design approach eliminates the need for rewriting C source code into HDL code and shortens both the development time and the simulation time.

When a design environment based on a C-based language is constructed, two problems occur that strongly depend on the high-level synthesis tool and the possibility for reusing properties from current and previous designs. Using a C-based language for the original design produces increasingly complex code, which often entails an increase in development time. To resolve these problems, we adopted a design technique based on the M language, which facilitates the construction of simulation environments.

The adoption of this method is expected to alleviate the difficulties associated with design based on C-based languages, which is implemented only by skilled software engineers, and design based on manual HDL coding, which is implemented only by skilled hardware engineers for embedded system design. This method allows both software engineers and hardware engineers to perform verification and debugging at the algorithmic level. It also allows for specifications to be promptly and accurately implemented as hardware modules by system designers and algorithm designers.

On the other hand, the conversion of floating-point operations into fixed-point operations is one of the important issues which must be resolved in order to ensure the seamless integration of all steps from algorithm development to hardware development. Therefore, in this work we intentionally adopted techniques based on the M language.

However, the support for automation for this language is not at a satisfactory level, whereas design based on C-based languages is 100% automated. Since the results of these trials include application-specific integrated circuit (ASIC) optimization, the same results or trends might not be observed when designing different types of hardware. However, it is clear that TAT can be consistently reduced. It is possible that without strong dependence on high level synthesis tools, M-based design can be regarded as a process of combining block modules from a database. We believe that such algorithm modules can accelerate and improve the reuse of properties from current and previous designs while providing the possibility for visualization of the design flow.

VII. CONCLUSION

Simulation environments for hardware design can be built in a straightforward manner by using M-based design, and simulations can be readily conducted in such environments. The M language was also found to improve the reusability of intellectual property from current and previous hardware designs. This design method supports automatic conversion from floating-point to fixed-point operations, which is highly useful in hardware design.

In addition, the design method can be used to analyze tradeoff tendencies in circuit performance by taking into account the input signal and the operating frequency. Therefore, M-based design can increase the productivity of embedded systems designers. In the design of complex embedded systems, this design method might prove highly effective for shortening the development period and reducing the incidence of serious flaws.

VIII. FUTURE WORK

Currently, the adoption of the M language for all hardware development is not possible since advanced functions, complex control logic, a complete environment for fulfilling equivalency checking, and formal property checking of automatically generated HDL code are as yet unavailable. We believe that when such functionality is implemented, hardware development time can be dramatically reduced and

cosimulation and co-verification can be performed by assigning equivalence relations between M-based modules and HDL modules. We plan to continue our investigations in this direction by using M-based design.

REFERENCES

[1] W.E. Leonard and W.S. Levine, "Using MATLAB to Analyze and Design Control Systems," Benjamin-Cummings Publishing, 1995.

[2] R.C. Gonzalez, R.E. Woods, and S.L. Eddins, "Digital Imaging Processing Using MATLAB," Prentice Hall Press, 2007.

[3] J.W. Woods, "Multidimensional Signal, Image, and Video Processing and Coding," 2nd edition, Academic Press 2011.

[4] O. Schliebusch, H. Meyr, and R. Leupers, "Optimized ASIP Synthesis from Architecture Description Language Models," Springer, pp.1-21, 2007.

[5] P. Coussy, G. Gajski, A. Takach, and M. Meredith, "An Introduction to High-Level Synthesis", Special issue on High Level Synthesis, IEEE Design and Test of Computers, Vol.26, No.4, 2009.

[6] G. Martin and G. Smith, "High-Level Synthesis: Past, Present, and Future," IEEE Design & Test of Computers, Vol.26, No.4, pp.18-25, 2009.

[7] Z. Zhang, Y. Fan, W. Jiang, G. Han, C. Yang, and J. Cong, "AutoPilot: A Platform-Based ESL Synthesis System," in HighLevel Synthesis: From Algorithm to Digital Circuit, Ed. P. Coussy, A. Morawiec., 2008.

[8] K. Wakabayashi, "C-based synthesis experiences with a behavior synthesizer, "Cyber"," in Proc. of DATE'99, pp.390393, 1999.

[9] B.K. Gunturk, J. Glotzbach, Y. Altunbasak, R.W. Schafer, and R.M. Mersereau, "Demosaicking: color filter array interpolation," IEEE Signal Processing Magazine, Vol.22, pp.44-54, 2005.

[10] J. Astola and P. Kuosmanen, "Fundamentals of Nonlinear Digital Filtering," Boca Raton, FL: CRC, 1997.

[11] P.P. Vaidyanathan, "Multirate Systems and Filter Banks," Prentice Hall 1993.

[12] Image web http://users.ecs.soton.ac.uk/msn/book/new_demo/ site:

[13] A. Matsuda and S. Baba, "An Automated Design Flow for Image Processing Filter in Embedded Systems," in Proc. of the 9th IEEE International Conference on ASIC (ASICON2011), Vol.1, pp.768-771, 2011.

[14] A. Matsuda and S. Baba, "Case Study of Filter Design in Embedded Systems," in Proc. of the 2011 IEEE Student Conference on Research and Development (SCOReD), pp.271276, 2011.

ISSN 2210-142X

Performance Evaluation of Dynamic Source Routing Protocol (DSR) on WSN

Hasein Issa Sigiuk1 and Ali A.S. Ihbeel2
1Electrical and Electronic Engineering Department, University of Tripoli, Tripoli, Libya sigiuk@mwc.ly
2Electrical and Electronic Engineering Department, University of Tripoli, Tripoli, Libya ali.ihbeel@yahoo.com

ABSTRACT

Wireless Sensor Network (WSN) has been regarded as a distinguished Ad Hoc Network that can be used for a specific application. Since a WSN consists of potentially hundreds of low cost, small size and battery powered sensor nodes, it has more potentials than others Ad Hoc networks to be deployed in many emerging areas. A number of routing protocols have been implemented to route the packets in these networks. One of these routing protocols is Dynamic Source Routing protocol (DSR). In this paper, an attempt has been made to evaluate the performance of DSR routing protocol using some simulation network models, to investigate how well this protocol performs on WSNs, in static and mobile environments, using NS-2 simulator. The performance study will focus on the impact of the network size, network density (up to 450 nodes), and the number of sources (data connections). The performance metrics used in this work are average end-to-end delay, packet delivery fraction, routing overheads, and average energy consumption per delivered packet.

Keywords: Wireless Sensor networks; Ad Hoc Networks; DSR Protocol; and Performance study.

1. INTRODUCTION

Wireless sensor network (WSN) is a wireless network consisting of small nodes with sensing, computation, and wireless communications capabilities [1]. The sensors measure ambient conditions in the environment surrounding and then transfer measurements into signals that can be processed to reveal some characteristics about phenomena located in the area around sensors [2]. However, sensor nodes are constrained in energy supply and bandwidth. Such constraints, combined with a typical deployment of large number of sensor nodes, have posed many challenges to the design and management of sensor networks [2]. Distinguished from traditional wireless communication networks, for example, cellular systems and Mobile Ad Hoc Networks (MANETs), WSNs have unique characteristics, for example, denser level of node deployment, higher unreliability of sensor nodes, and severe energy, computation, and storage constraints [3], which present many new challenges in the development and application of WSNs. WSN may also have some interesting features including self-organization, dynamic network topology, and multi-hop routing, which are important for many real world applications, cover many areas such as: disaster management, border protection, combat field surveillance, and any place where humans cannot easily access or unsafe to human life [4].

Although sensor networks and MANETs are similar to some extent, they are radically distinct in many aspects, both MANETs and WSNs belong to Ad Hoc networks and built on top of wireless communication channels; nodes communicate with each other through multi-hop links; each node serves as a router to forward packets for others; and nodes are resourceconstrained and usually powered by batteries. The differences include that: the sensor nodes are usually densely deployed in a field of

of interest and the number of them can be several orders of magnitude higher than that in a MANET; severe energy, computation, and storage constraints; sensor networks are application specific, and usually designed and deployed for a specific application; network topology changes frequently due to node failure, damage, or energy depletion; and in most sensor network applications, the data sensed by sensor nodes flow from multiple source sensor nodes to a particular sink, exhibiting a many -to- one traffic pattern [1, 2, 3, 4].

There are many routing protocols which have been proposed for Ad Hoc networks. These include: Ad hoc on Demand Distance Vector (AODV), Dynamic Source Routing (DSR), Destination Sequenced Distance Vector (DSDV), Temporally Ordered Routing Algorithm TORA and Optimized Link State Routing (OLSR). These protocols have been investigated on the MANETs in the past few years [7, 8, 9]. The Performance investigation of these protocols, on the MANETs, has produced many useful results. We have seen very limited findings on how these Ad hoc routing protocols perform on WSNs [4, 5, 6, 11, 12].

The objective of this research is to carry out a systematic performance study on DSR, and in particular its capability to be used as a routing protocol on WSNs, as it was originally designed to be used in MANETs. In this paper a performance study is to be conducted for DSR protocol, where some specific parameters, such as network size, network density (up to 450 nodes), and number of sources (data connections) are considered and investigated for their effect on the performance of WSN.

The rest of the paper is organized as follows: Section II includes the recent related works. The DSR routing protocol description is summarized in section III. The simulation environment and performance metrics are described in Section IV. The experimental results are presented in section V. The paper is concluded in section VI.

II. RELATED WORKS

Z.Zhang and et al [4] investigate how well Ad Hoc routing protocols work on WSNs with a different number of sources. Average end-to-end delay, packet delivery fraction and routing overheads were examined for 50 nodes in (1500×300) m2 network for five routing protocols namely AODV, DSR, DSDV, TORA and OLSR. The simulation study carried out for these routing protocols, using different scenarios, showed that there are some merits and drawbacks. The performance comparison of these routing protocols showed that the AODV always was performing better on all WSN tested models with single or multiple sources. The DSDV was next to the AODV despite of the relatively low packet delivery fraction of the DSDV. M. N. Jambli and et al [5] evaluated the capability of AODV on how far it can react to network topology change in Mobile WSN. They investigated the performance metrics namely packet loss and energy consumption of mobile nodes with various speed, density and route update interval (RUI), for 9 nodes in (100×100) m2 network. The presented results showed a high percentage of packet loss and the reduction in total network energy consumption of mobile nodes if RUI is getting longer due to serious broken link caused by nodes movement.

M. Pandey and et al [6] presented an analytical study of the average jitter of AODV Routing protocol in wireless sensor networks, for different simulation time and mobility conditions. The performance measurements were carried out for the AODV routing protocol for different simulation times and network topologies and under different mobility conditions. The paper investigated the impact of different mobility models on the performance of 105 nodes in (500×500) m2 wireless sensor networks. Although the presented results did not present a steep comparative orientation of the results towards a specific routing protocol but the comparative study leads towards some interesting results.

Peter Kok and et al [11] performed some simulation experiments using EAR, Gradient Broadcast (GRAB), Gradient Based Routing (GBR), DSR, and AODV routing protocols,in terms of packet

latency, packet delivery fraction and average energy consumption per delivered packet. The experimented simulation models consist of 400 nodes with 10% and 50% active source nodes. The simulation results demonstrated that a routing protocol which its design based on a combination of routing parameters exhibits collectively better than other protocols with their design based on just hop count and energy or those using flooding.

III. DYNAMIC SOURCE ROUTING PROTOCOL (DSR)

The distinguishing features of DSR are: low network overhead, no extra infrastructure for administration and the use of source routing. Source routing implies that the sender had full knowledge of the complete hop-by-hop route information to the destination. The protocol is composed of the two main mechanisms of Route Discovery and Route Maintenance. Normally routes are stored in a route cache of each node. When a node likes to communicate to a destination, first it checks for the route for that particular destination in the route cache. If yes, the packets are sent with source route header information to the destination. In the other case, if the route is not available at the route cache; then the node will initiate the route discovery mechanism to get the route first. The route discovery mechanism will flood the network with route request (RREQ) packets, and then the neighbors will receive RREQ packets and check for the route to destination in their route cache. If the route is not in their caches rebroadcast the RREQ, otherwise the node replies to the originator with a route reply (RREP) packet. Since RREQ and RREP packets both are source routed, original source can obtain the route and add to its route cache. In any case the link on a source route is broken; the source node is notified with a route error (RERR) packet. Once the RERR is received, the source removes the route from its cache and route discovery process is reinitiated [13].

DSR being a reactive routing protocol has no need to periodically flood the network for updating the routing tables like table-driven routing protocols do. Intermediate nodes are able to utilize the route cache information efficiently to reduce the control overhead.

IV. SIMULATION ENVIRONMENT & SETUP

A. Simulation Model

In our experiments we use NS-2 (version 2.32), a discrete event simulator widely used in the networking research community, as a flexible tool for networking researchers to investigate how various routing protocols perform with different network configurations and topologies [14]. NS-2 simulator was validated in [10] and verified in a number of later publications, e.g. [8]. There are two scenarios, Static, and Mobile.The wireless sensor network application under consideration in this work is environmental data collection wireless sensor network, i.e. is one of WSN applications [1]. In this application, large numbers of sensors are deployed in the field to measure different parameters such as temperature, speed, humidity and direction. In data collection applications the sensor nodes remains sleep most of the time and report measurements frequently to the base station (sink). The deployment of large scale sensors in such applications either static or mobile and they may be equipped with effective power scavenging methods, such as solar cells [1]. In some other applications, sensors are mounted on robots, animals or other moving objects, which can sense and collect relevant information [5]. In the simulation, source nodes, (The source nodes are the sensor nodes that have detected phenomena and need to transmit the sensed data to the sink node) generate data packets that are routed to the sink located in the center of the WSN, the source nodes follow a Gaussian distribution in generating packets.

To allow comparison with other experiments [4, 5, 6, 8], we use 512 byte data packets and CBR traffic. For the impact of network size we simulate populations of 100, 200, 300, 400, and 450 nodes in areas of 2121m×425m, 3000m×600m, 3675m×735m, 4250m×850m, and 5000m×900m for 200s of simulation

time with 10 CBR sources. We choose the above combinations of areas and number of nodes involved to work with approximately the same node density and simulation area proportions. This density of nodes is high enough to allow a meaningful comparison of the protocols; a markedly lower density may cause the network to be frequently disconnected, and then an investigation of the efficiency of different routing protocols is even more complicated. For the impact of network density (populations), we simulate populations of 100, 200, 300, 400, and 450 nodes in an area of 2125m×2125m with 10 CBR sources. In addition to that, we simulate 10%, 20%, 30%, 40%, and 50% CBR traffic sources for 2121m × 425m network size with 100 nodes. This number of sources allows us to investigate scalability of protocols when the traffic load is changed from light load to heavy load. Such settings is more realistic for WSNs. All peer-to-peer connections are started at times uniformly distributed between 0 and 100s. The number of unique traffic sources is 70% of the total number of sources. The chosen sending rate is 4 packets/s. Each data point presented in this paper is an average of five runs, each lasting for 200 s of simulated time. The IEEE 802.11 Distributed Coordination Function (DCF) is used as the Medium Access Control Protocol with the suggested parameters to model 914MHzLucentWaveLAN DSSS radio interface at a 2 Mb/s data rate. The adjusted parameters in the simulation are given in table.1.

TABLE.1: PARAMETERS USED IN THE SIMULATION

Parameter	Mobile Scenario	Static Scenario
Max. number of nodes (N)	450 nodes	450 nodes

TABLE.1: PARAMETERS USED IN THE SIMULATION

Parameter	Mobile Scenario	Static Scenario
MAC type	IEEE 802.11/ DCF	IEEE 802.11/ DCF
Propagation model	Two ray ground	Two ray ground
Traffic type	Constant bit rate	Constant bit rate
Agent	UDP	UDP
Queue length	50 packets	50 packets
Connection Rate	4 pkts/sec	4 pkts/sec
Tx power	0.2818 W	0.2818 W
Transmission range	250 m	250 m
Initial energy	200J	200J
Simulation time	200 seconds	200 seconds
Node mobility	Random waypoint	NA
Pause time	50 sec	NA
Max speed of mobile node	5 m/sec	NA

B. Performance Metrics

The evaluation is done using the following metrics:

a) Packet Delivery Fraction (PDF): measures the percentage of data packets generated by nodes that are successfully delivered to the sink, expressed as: (Total number of data packets successfully delivered)/ (Total number of data packets sent) ×100%

b) Average End-to-End Delay: measures the average time it takes to route a data packet from the source node to the sink. It is expressed as: (∑ Individual data packet latency)/ (∑ Total number of data packets delivered)

c) Routing Overheads (ROH): The average number of control packets produced per sensor node. The control packets include route requests, replies and error messages.

d) Energy Consumption per Delivered Packet: This measures the energy expended per delivered data packet. It is expressed as: (∑ Energy expended by each node)/ (Total number of delivered data packets)

V. SIMULATION RESULTS

A. Impact of the number of nodes

The density of nodes expected to have a significant influence on the performance of DSR. Low density may cause the network to be frequently disconnected. High density increases the contention. This experiment shows the effect of changing the node density (number of nodes in the network) on the performance of the DSR protocol. Figure 1 depicts the PDF, average end-to-end delay, ROH, and average energy consumption per delivered packets measured with 100, 200, 300, 400, and 450 nodes for static and mobile deployment scenarios. In terms of PDF, it is found that, for both static and mobile nodes, the performance of the DSR is slightly degraded at low node density. However, as the reliability of the network is improved with the increase in the

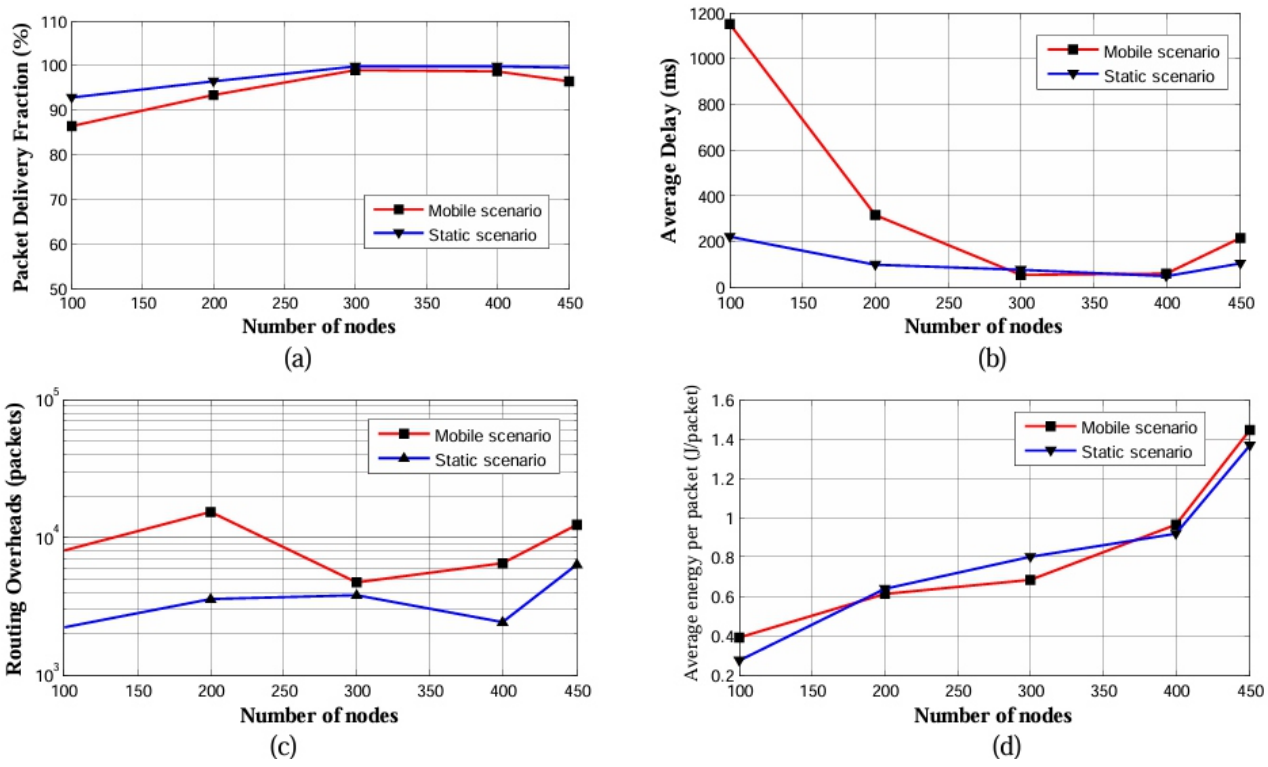


Figure.1: Impact of the number of nodes.

number of nodes, it is noticed that the PDF for static node deployment performs better as the node density increase, as shown in figure 1-(a). As the DSR routing protocol caches all known routes; it is very likely that during route discovery for some destination such as node D, a route for another node A is found, recorded, and later used from the cache, this strategy will ultimately save the network bandwidth, which leads to improve the performance of DSR protocol, especially when the number of nodes increase.

From figure 1-(b), it is noticed that, for both static and mobile deployments, the best average end-to-end delay exhibited by DSR when the number of nodes in the network are between 300 and 400; it is almost less than 0.1 s. However, the average end-to-end delay is found to be degraded as the node density decreases, especially for mobile node deployment case. In terms of ROH, as shown in figure 1-©, it is noticed that the DSR protocol generates a higher routing load for mobile nodes deployment. There is minor increase in the routing overheads as the number of nodes increases.

As shown in figure 1-(d), the Average energy consumption increases as the number of nodes in the network increases.

B. Impact of the network size

In this experiment we study the performance of the DSR protocol in areas of 2121m × 425m, 3000m × 600m, 3675m × 735m, 4250m × 850m, and 5000m × 900m populated by 100, 200, 300, 400, and 450 nodes, respectively. It is worth to mention that for all the above combinations of areas and nodes, the density of nodes is kept constant. Figure 2 shows the simulation results for this experiment. In terms of PDF, as shown in figure 2-(a), the DSR performs well with the changes made in the network size. However its performance declines beyond 200 nodes for mobile scenario. It is noticed that the DSR managed to deliver more than 95% in small size networks (less than 200 nodes), however, for larger networks the performance declines. Generally, for both scenarios, it is noticed that as the number of nodes grows beyond 200 nodes, (the network size increased), the PDF starts to decline.

In terms of end-to-end delay both static and mobile scenarios exhibit in a similar fashion, for small sized networks, as shown in figure 2-(b).However, as the number of nodes grows, for larger networks (more than 200 nodes), there will be a noticeable degradation in the end-to-end delay performance, especially for the mobile scenario. Figure 2-© shows that the DSR protocol has demonstrated significant lower routing overheads for the static scenario, in comparison to that of the mobile scenario. It is noticed that the overhead increases as the network size becomes larger.

According to the results presented in figure 2-(d), for the average consumed energy per packet, it is noticed that the DSR protocol has demonstrated a remarkable performance with lower energy consumption for small sized networks. The consumed energy increases as the network size increases.

C. Impact of the number of sources

Figure.3 depicts the effects of network loading on the performance of the DSR protocol by increasing the number of data connections (number of sources), from 10% to 50% for 100 nodes in the network. Figure 3-(a) shows that the PDF declines with the increase in the number of active sources for both scenarios. The average end-to-end delay performance of both scenarios, as shown in figure 3-(b), degrades as the number of connections increases in the network, and with the mobile scenario shows more sharp increase in the delay. The decline in the network average latency performance results for the DSR protocol appears to coincide with the performance results obtained for the PDF experiment.

The results presented in figure 3-© show that the DSR protocol has demonstrated a lower ROH for light traffic in the network. As the number of connections increases, the ROH performance results are almost moderate and consistent.

Figure 3-(d) shows that the DSR protocol consumes more energy for a heavy traffic network, and the results for average energy consumption appear to exhibit in a similar fashion to that experienced in the ROH performance test.

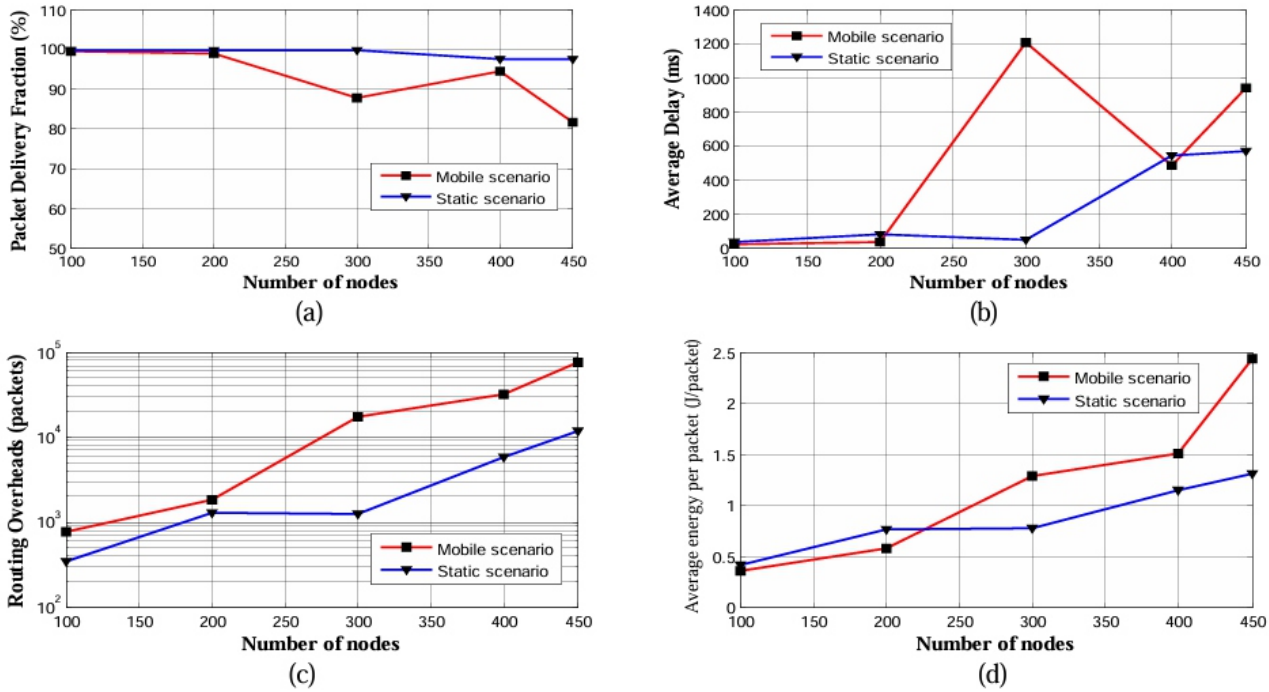


Figure.2: Impact of the network size.

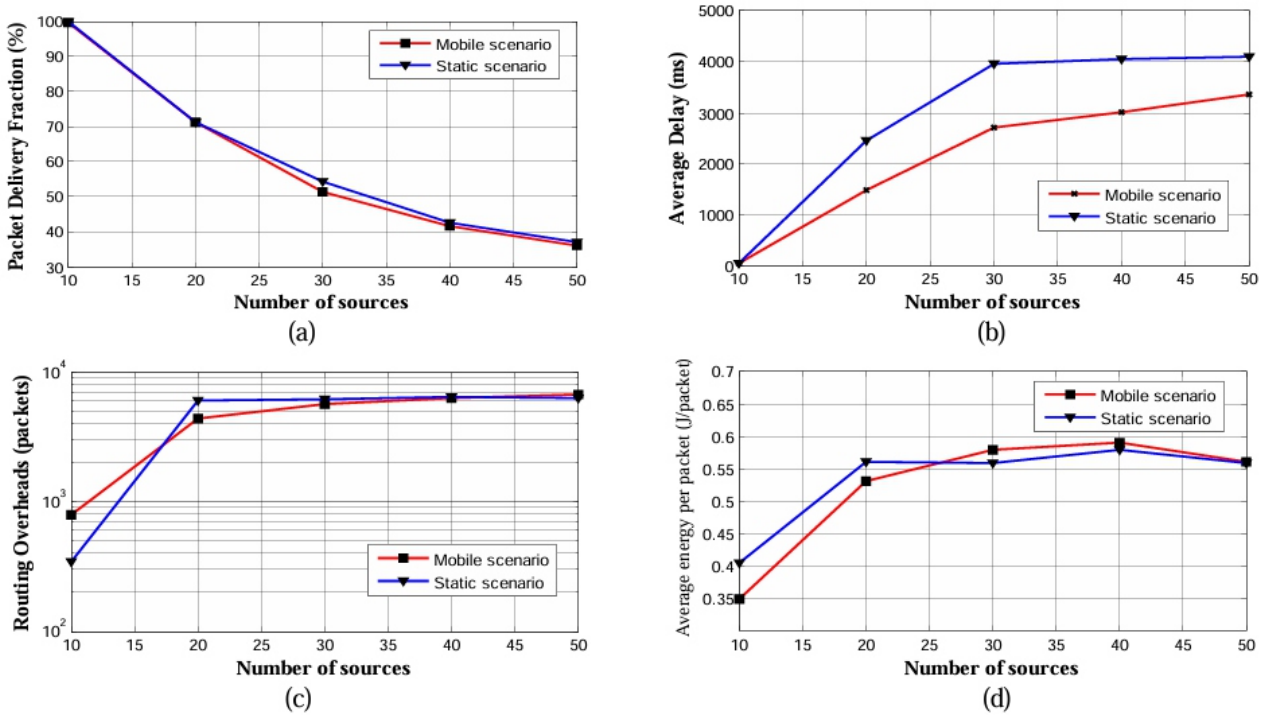


Figure.3: Impact of the number of sources.

This indicates that the energy consumed is proportional to the ROH in the network. Generally, as one can see from the presented experimental results for the DSR protocol, it is noticed that the performance of WSN degrades as the number of connections in the network increases.

VI. Conclusions

In this paper, we have presented a performance evaluation for DSR protocol under various WSN scenarios using different performance metrics which are the average end-to-end delay, packet delivery fraction, routing overhead and the energy consumption per delivered packets, with the impact of the network size, network density, and the number of sources.

It is found that in most of the tested scenarios, the DSR protocol performs well and its performance is better for the static scenario application. Because of the multiple paths that are already registered and kept in the route cache of the nodes, where a good degree of reliability and stability is provided by the network. The DSR protocol exhibits high PDF, low latency and energy consumption, and managed to adapt to the changes in the network like density and size. Under heavy load conditions, it is noticed that there is performance degradation, in terms of PDF and delay. The work presented here aims to find out the effect of different parameters on the performance of the DSR routing protocol in WSN. The results though don't present a steep comparative orientation of the results towards a specific routing protocol, but the comparative study leads towards some interesting results. Further research is needed to find out the most suitable protocol for each application.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Communications Magazine, pp. 102–114, August, 2002.

[2] J. Alkarake and A. Kamal, "Routing techniques in wireless sensor networks," A survey, IEEE wireless communications, pp. 6-27, December, 2004.

[3] J. Zheng and A. Jamalipour, "Wireless Sensor Networks A Networking Perspective," by Institute of Electrical and Electronics Engineers, first edition, 2009.

[4] Z. Zhang, H. Zhou, and J. Gao, "Scrutinizing Performance of Ad Hoc Routing Protocols on Wireless Sensor Networks," First Asian Conference on Intelligent Information and Database Systems, pp. 459-464, IEEE 2009.

[5] M. N. Jambli, K. Zen, H. Lenando and A. Tully, "Performance Evaluation of AODV Routing Protocol for Mobile Wireless Sensor Network," 7th International Conference on IT in Asia (CITA), IEEE 2011.

[6] M. Pandey and S. Verma, "Performance Evaluation of AODV for different Mobility Conditions in WSN," International Conference on Multimedia, Signal Processing and Communication Technologies, pp. 240-243, IEEE 2011.

[7] N. Vetrivelan and A. V. Reddy, "Performance Analysis of Three Routing Protocols for Varying MANET Size," Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol. 2, March, 2008. Last accesed in March 2012.

[8] V. Naumov, T. Gross, "Scalability of routing methods in ad hoc networks," Elsevier Performance Evaluation Journal, Vol. 62, 2005.

[9] S. Gowrishankar, T.G. Basavaraju, M. Singh and S. Sarkar, "Scenario based Performance Analysis of AODV and OLSR in Mobile Ad hoc Networks," Proceedings of the 24th South East Asia Regional Computer Conference, November, 2007. Last accesed in March 2012.

[10] D. Johnson, "Validation of wireless and mobile network models and simulation," DARPA/NIST NSV Workshop, 1999.

[11] P. Loh, H. Jing and Y. Pan, "Performance Evaluation of Efficient and Reliable Routing Protocols

for Fixed-Power Sensor Networks," IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, Vol. 8, No. 5, 2009.

[12] F. Ullah, M. Amin and H. Ghaffar, "Simulating AODV and DSDV For A dynamic Wireless Sensor Networks," International Journal of Computer Science and Network Security, Vol.10, No.7, pp. 219-223, July, 2010. Last accesed in January 2012.

[13] J. Broch, D. Maltz, D. Johnson, Y. Hu and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," Proceedings of the MOBICOM'98, pp. 8597, 1998.

[14] "The Network Simulator – ns-2", available at: www.isi.edu/nsnam/ns. Last accessed in March 2012.

ISSN 2210-142X

Finite Element Approach of Unshielded Multiconductor Transmission Lines Embedded in Layered Dielectric Region for VLSI Circuits

Sarhan M. Musa and Matthew N. O. Sadiku
Roy G. Perry College of Engineering, Prairie View A&M University, Texas 77446, USA smmusa@pvamu.edu

ABSTRACT

This paper presents the quasi-TEM approach for the analysis of unshielded multiconductor transmission lines interconnect in single and two-layered dielectric region using the finite element method (FEM). We mainly focus on designing of four-transmission lines embedded in two-layered dielectric media and five-transmission lines interconnect in single-layered dielectric medium. We compute the capacitance matrices for these configurations. Also, we determine the quasi-TEM spectral for the potential distribution of the integrated circuits. .

Keywords: Capacitance per unit length; fnite element method; VLSI; multiconductor; transmssion lines.

1. INTRODUCTION

In The optimization of the electrical performance of microelectronic integrated circuits becomes an important factor when the signal speeds and components densities increase. Therefore, the accurate and efficient computational of self and mutual (coupled) capacitances of multicondcutor interconnect in very high speed integrated circuits is essential for scientists and researchers.

The characteristics of microwave integrated circuits' analysis and design must be accomplished accurately in a short time. The significant advantages of printed circuits are somewhat offset by the electromagnetic complexity of the structure, because its inherent inhomogeneous nature makes accurate calculations difficult.

Several methods used for analyzing multiconductor transmission lines include the method of moments (MoM) [13], the measured equation of invariance (MEI) [4-6], the Fourier projection method (FPM) [7], the matrix pencil method (MPM) [8], the Fourier transform and mode-matching techniques (FTMM) [9], the partial element equivalent circuit methods (PEEC) [10], the method of line (MoL) [11], the spectral domain analysis (SDA) [12], and the finite difference method [13].

The quasi-TEM approach is successfully used to analyze coupled-microstrip lines of finite length in [14]. Indeed, the problem of computing the capacitance coupling in Very Large Scale Integrated (VLSI) circuits is studied using an approximate extended version of the method of images [15]. Also, the capacitance of a conductor-backed coplanar waveguide with an upper shielding was investigated in [16]. Although, the Eigenmode-based capacitance calculations with applications in passivation layer design was investigated in [17].

We use FEM in designing the four-transmission lines interconnect in two-layered dielectric media and fivetransmission lines interconnect in single-layered dielectric medium structures. The FEM is especially suitable and effective for the computation of electromagnetic fields in strongly inhomogeneous media. Also, it has high computation accuracy and fast computation speed. We show that FEM is as suitable and effective as other methods for modeling multiconductor transmission lines of the VLSI circuits.

We compared some of our results of computing the capacitance-per-unit length with the other methods. We specifically compared the modeling of designing of unshielded four-transmission lines interconnect in two-layered dielectric media with the method of moments, measured equation of invariance, and Fourier projection method. Also, results from the matrix pencil method, and Fourier transform and modematching techniques were compared for unshielded fivetransmission lines interconnect in single-layered dielectric medium and found to be in agreement.

II. THEORY FOR THE PROBLEM FORMULATION OF MULTICONDUCTOR INTERCONNECTS IN MULTILAYERED DIELECTRIC MEDIA

In The models are designed in 2D using electrostatic environment in order to compare our results with some of the other available methods. In the boundary condition of the model's design, we use ground boundary which is zero potential (0 V) for the shield. We use port condition for the conductors to force the potential or current to one or zero depending on the setting. Also, we use continuity boundary condition between the conductors and between the conductors and left and right grounds.

The quasi-static models are computed in form of electromagnetic simulations using partial differential equations. Recently, with the advent of integrated circuit technology, the coupled microstrip transmission lines consisting of multiple conductors embedded in a multilayer dielectric medium have led to a new class of microwave networks. Multiconductor transmission lines have been utilized as filters in microwave region which make it interesting in various circuit components. For coupled multiconductor microstrip lines, it is convenient to write:

$$Q_i = \sum_{j=1}^{m} C_{sij} V_j \ , (i = 1, 2, ..., m) \ , \qquad (1)$$

where *Qi* is the charge per unit length, *Vj* is the voltage of *Jth* conductor with reference to the ground plane, *Csij* is the short circuit capacitance between *i th* conductor and *jth* conductor. The short circuit capacitances can be obtained either from measurement or from numerical computation. From the short circuit capacitances, we obtain

$$C_{ii} = \sum_{j=1}^{m} C_{sij} \ , \qquad (2)$$

where *Cii* is the capacitance per unit length between the *ith* conductor and the ground plane. Also,

$$C_{ij} = -C_{sij}, \qquad j \neq i, \qquad (3)$$

where *Cij* is the coupling capacitance per unit length between the *i th* conductor and *j th* conductor. The coupling capacitances are illustrated in Fig. 1.

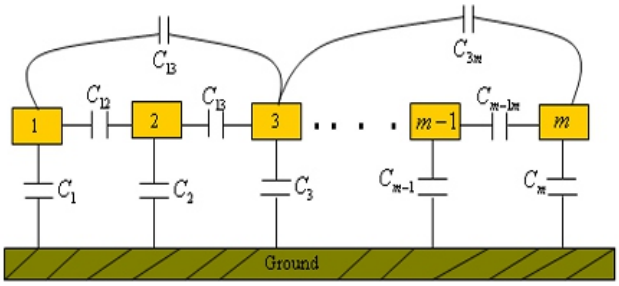


Figure 1. The per-unit length capacitances of a general m-conductor transmission line.

For m-strip line, the per-unit-length capacitance matrix [C] is given by [18]:

$$[C] = \begin{bmatrix} C_{11} & -C_{12} & \cdots & -C_{1m} \\ -C_{21} & C_{22} & \cdots & -C_{2m} \\ \vdots & \vdots & & \vdots \\ -C_{m1} & -C_{m2} & \cdots & C_{mm} \end{bmatrix}. \qquad (4)$$

III. RESULTS AND DISCUSSION

The models are designed with finite elements are unbounded (unshielded or open), meaning that the electromagnetic fields should extend towards infinity. This is not possible because it would require a very large mesh. The easiest approach is just to extend the simulation domain "far enough" that the influence of the terminating boundary conditions at the far end becomes negligible. In any electromagnetic field analysis, the placement of far-field boundary is an important concern, especially when dealing with the finite element analysis of structures which are open. It is necessary to take into account the natural boundary of a line at infinity and the presence of remote objects and their potential influence on the field shape [19]. In all our simulations, the open multiconductor structure is surrounded by a W X H shield, where W is the width and H is the thickness.

The models are designed in two-dimensional (2D) using electrostatic environment in order to compare our results with the other available methods. In the boundary condition of the model's design, we use ground boundary which is zero potential (0 V) for the shield. We use port condition for the conductors to force the potential or current to one or zero depending on the setting. Also, we use continuity boundary condition between the conductors and between the conductors and left and right grounds. The quasi-static models are computed in form of electromagnetic simulations using partial differential equations.

In this paper, we consider two different models. Case A investigates the designing of unshielded four-transmission lines interconnect in two-layered dielectric media. For case B, we illustrate the modeling of unshielded five-transmission lines interconnect in single-layered dielectric medium. The results from both models are compared with some other results in the other methods and found to be close. A. Unshielded Four-Conductor Transmission Lines Figure 2 shows the cross section for unshielded four conductor transmission lines with the following parameters:

ε_{r1} = dielectric constant of the dielectric material = 5.0.

ε_{r2} = dielectric constant of the free space = 1.0.

W = width of the dielectric material = 10mm.

w = width of a single conductor line = 1mm.

H_1 = distance of conductors 1 and 2 from the ground plane = 3mm.

H_2 = distance of conductor 4 from the ground plane = 1mm.

H_3 = distance of conductor 3 from the ground plane = 2mm.

s = distance between the two coupled conductors = 1mm.

t = thickness of the strips = 0.01mm.

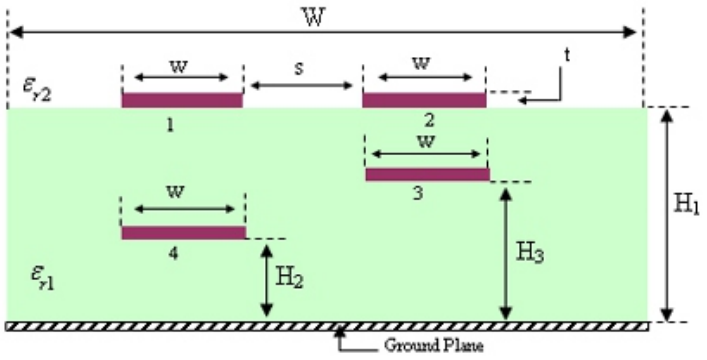


Figure 2. Cross section of unshielded four-conductor transmission lines.

The geometry is enclosed by a 10 X 10mm shield. From the model, we generate the finite elements mesh as in Fig. 3. Table 1 shows the statistical properties of the mesh. While, Fig. 4 shows the contour plot of the potential distribution with port 1 as input. The potential distribution along the line that goes from (x,y) = (0,0) to (x,y) = (10mm, 10mm) is portrayed in Fig. 5.

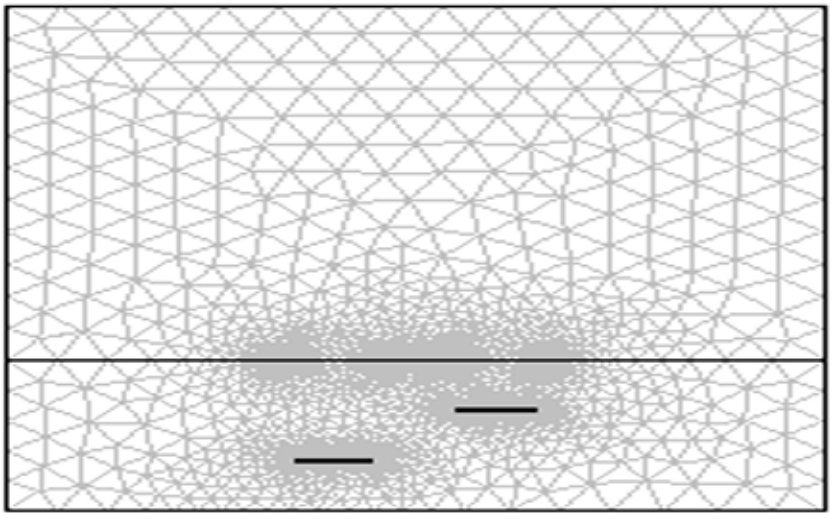


Figure 3. Mesh plot of unshielded four-conductor transmission lines.

TABLE I. MESH STATISTICS OF THE UNSHIELDED FOUR-CONDUCTOR TRANSMSSION LINES

Items	Value
Number of degrees of freedom	103940
Total Number of mesh points	4731
Total Number of elements	9315
Triangular elements	9315
Quadrilateral	0
Boundary elements	548
Vertex elements	22

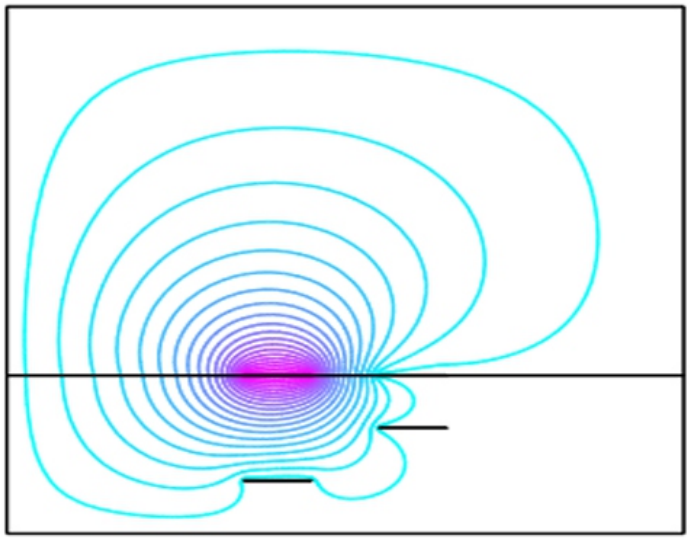


Figure 4. Contour plot of the potential distribution of unshileded four conductor transmission lines.

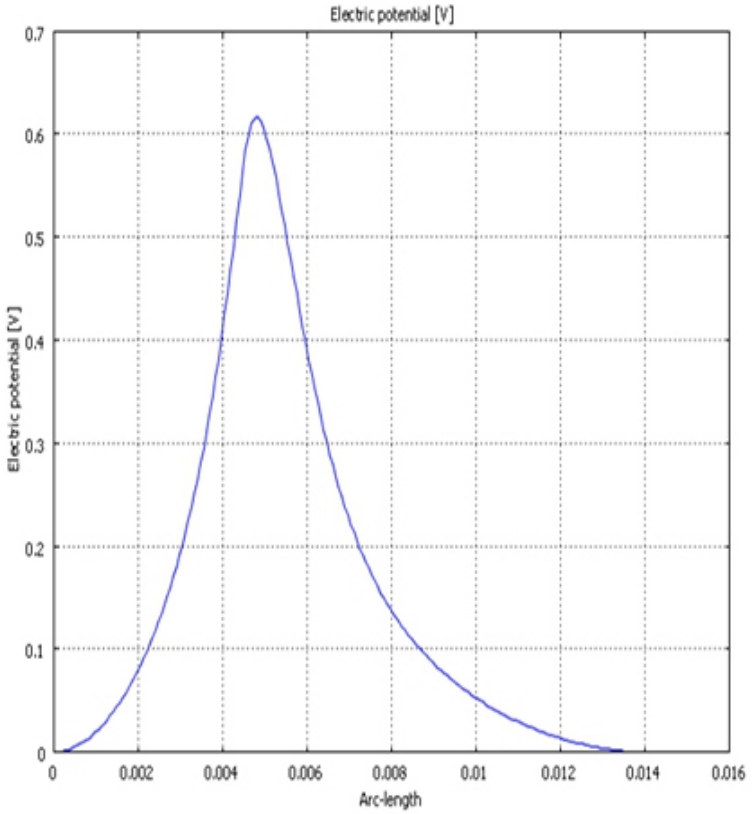


Figure 5. Potential distribution of unshielded four- conductor transmission lines using port 1 as input along a line from (x,y) = (0,0) to (x,y) =(10, 10) mm.

Table 2 shows the finite element results for the capacitance per-unit length of unshielded four-transmission lines embedded in two-layered dielectric media. It compares the results based on our work with those from other methods.

TABLE II. VALUES OF CAPACITANCE MATRIX (IN PF/M) FOR UNSHIELDED FOUR-CONDUCTOR TRANSMSSION LINES

Capacitance per unit length	MoM[1-3]	MEI[4]	FPM[7]	This work
C_{11}	70.158	89.514	70.158	73.052
C_{12}	-12.842	-12.832	-12.839	-12.948
C_{13}	-12.960	-13.110	-12.967	-13.239
C_{14}	-22.240	-23.014	-22.230	-22.549
C_{22}	87.327	87.028	87.227	90.823
C_{23}	-54.195	-55.462	-54.234	-56.029
C_{24}	-4.052	-3.988	-4.049	-3.924
C_{33}	133.935	128.861	128500	139.354
C_{34}	-15.606	-14.935	-14.210	-16.520
C_{44}	141.170	141.312	135.940	145.967

B. Unshielded Five-Conductor Transmission Lines

Figure 6 shows the cross section for the unshielded five conductor transmission lines with the following parameters:

ε_{r1} = dielectric constant of the dielectric material = 2.0.

ε_{r2} = dielectric constant of the free space = 1.0.

W = width of the dielectric material = 31mm.

w = width of a single conductor line = 3mm.

h = distance of conductors from the ground plane = 1mm.

s = distance between the two coupled conductors = 2mm.

t = thickness of the strips = 0.01mm.

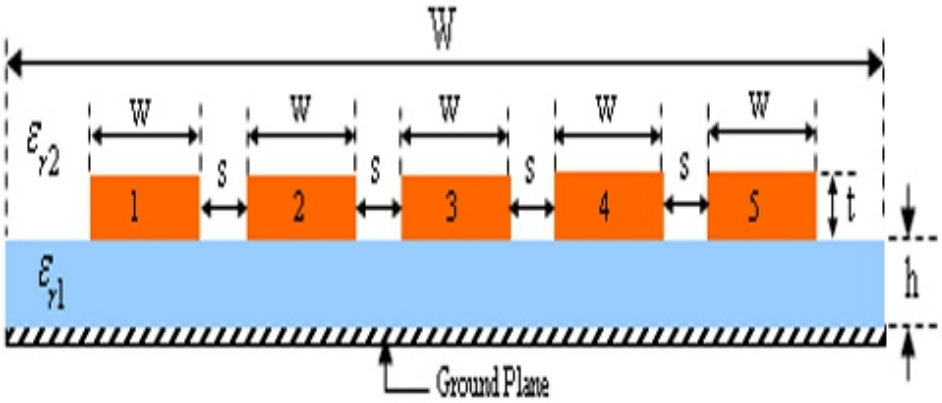


Figure 6. Cross section of unshielded five-conductor transmission lines.

The geometry is enclosed by a 31 X 5mm shield. Figure 7 shows the 2D surface distribution of the model. From the model, we generate the finite elements mesh plot as in Fig. 8. Table 3 shows the statistical properties of the mesh. While, Fig. 9 shows the contour plot of the potential distribution with port 1 as input. In addition, Fig. 10 shows the comparison analysis of the potential distribution of the model with and without dielectric substrate along the line that goes from (x,y) = (0,1mm) to (x,y) = (43mm, 1 mm). It observed that the peak valu of electric potential stayed same as the dielectric is placed in the substrate when we use the first conductor as input.

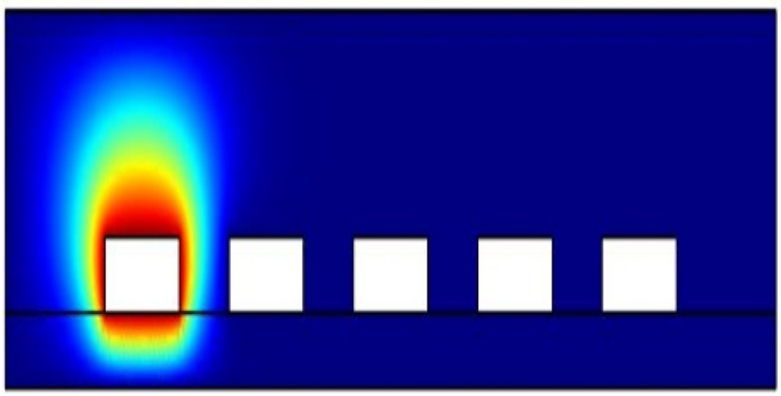


Figure 7. 2D surface potential distribution of the unshielded five-conductor transmission lines.

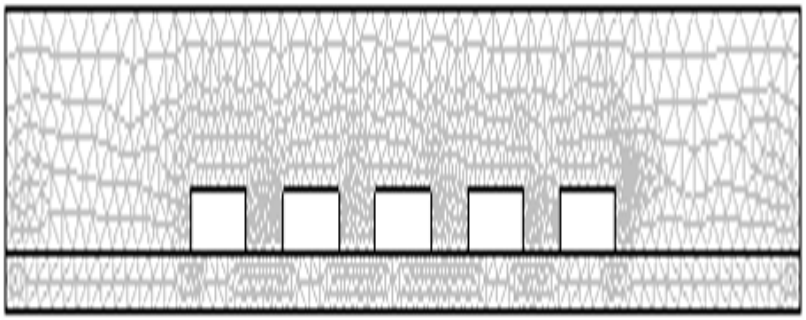


Figure 8. Mesh plot of the unshielded five-conductor transmission lines.

TABLE III. MESH STATISTICS OF THE UNSHIELDED FIVE-CONDUCTOR TRANSMSSION LINES

Items	Value
Number of degrees of freedom	21044
Total Number of mesh points	1004
Total Number of elements	1792
Triangular elements	1792
Quadrilateral	0
Boundary elements	272
Vertex elements	26

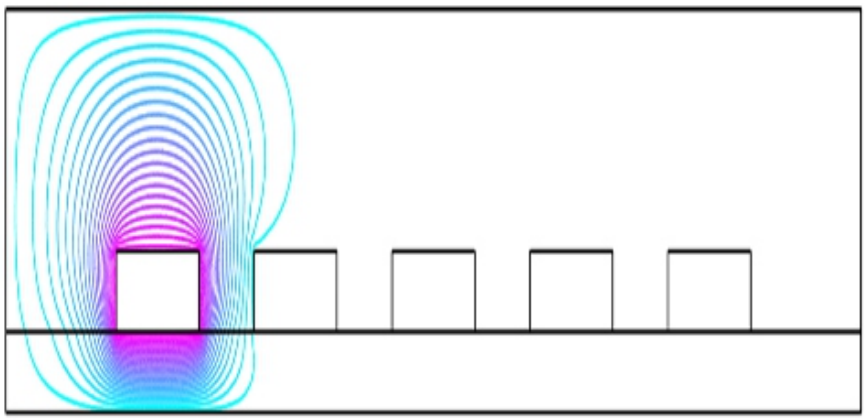


Figure 9. Contour plot of the potential distribution of the unshielded five conductor transmission lines.

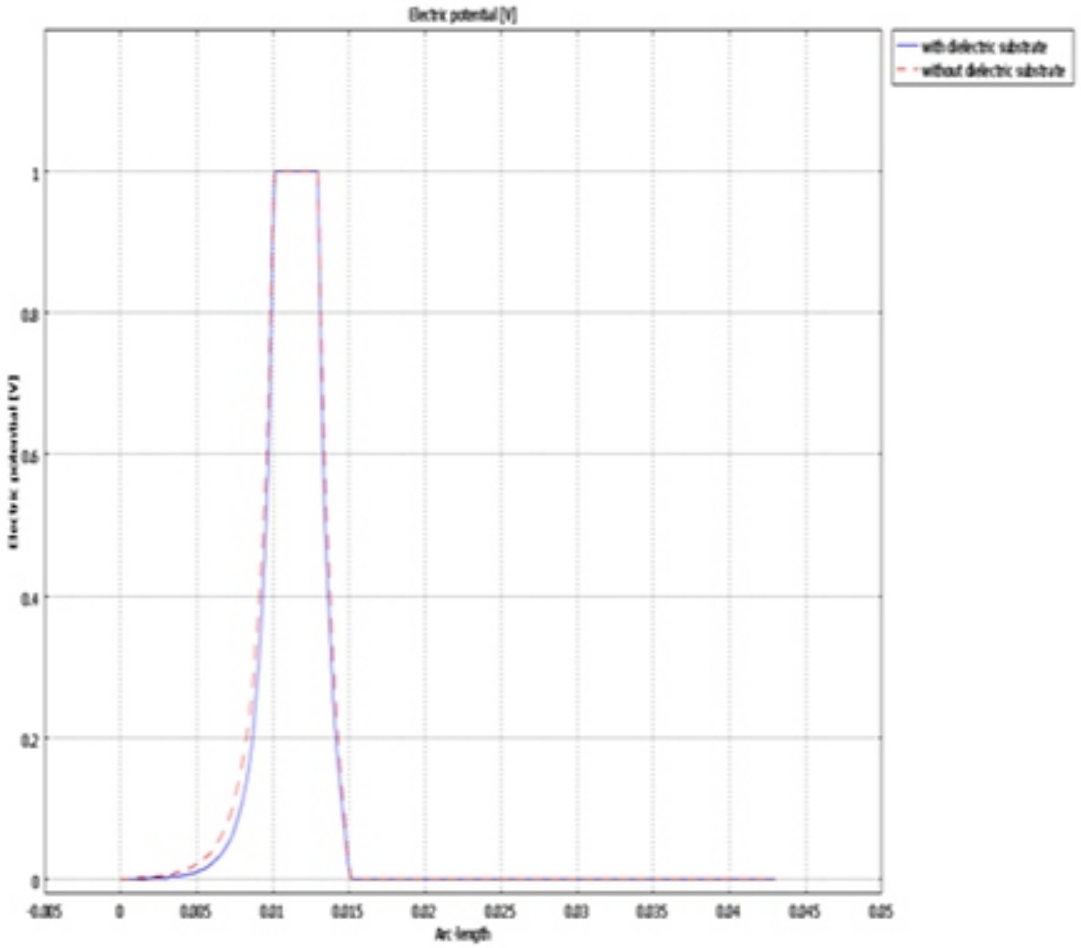


Figure 10. Comparison Analysis of Potential distribution of the model with and without dielectric substrate

Table 4 shows the finite element results for the capacitance per-unit length of the unshielded five-transmission lines interconnect in single-layered dielectric medium. It compares the results from our work with those from other methods.

TABLE IV. VALUES OF THE FELF AND MUTUAL CAPACITANCES COEFFICIENT (IN PF/M) FOR UNSHIELDED FIVE-CONDUCTOR TRANSMSSION LINES

Capacitance per unit length	(MPM) [8]	(FTMM) [9]	This work
C_{11}	93.668	89.660	100.073
C_{12}	-8.453	-8.110	-5.983
C_{13}	-0.809	-0.795	-0.016
C_{21}	-0.345	-0.319	-5.983
C_{22}	95.329	92.173	100.899
C_{23}	-8.318	-7.962	-5.975
C_{24}	-0.758	-0.730	-0.016
C_{33}	95.341	92.145	100.816

Tables 2 and 4 provide the results of FEM in two dimensional compared with other methods for the characteristics of unshielded two-layered multiconductor transmission lines and unshielded a single-layered multiconductor transmission lines with a thin strip thickness, respectively. The results of capacitance matrices for self and mutual capacitances, which are useful for the analysis of crosstalk between high-speed signal traces on the printed circuit board, are compared with other methods' data for the validity of the proposed method.

IV. CONCLUSION

In this article, we have presented the modeling in 2D of unshielded four-transmission lines embedded in two-layered dielectric media and five-transmission lines interconnect in single-layered dielectric medium. We have shown that FEM is suitable and effective as other methods for modeling multiconductor transmission lines in VLSI circuits. Some of the results obtained using FEM for the capacitance-per-unit length agree well with those found in other methods. Also, we determine the quasi-TEM spectral for the potential distribution of the models. The results obtained in this research are encouraging and motivating for further study.

REFERENCES

[1] C. Wei, R. F. Harrington, J. R. Mautz, and T. K. Sarkar, "Multiconductor transmission lines in multilayered dielectric media," IEEE Transactions on Microwave Theory and Techniques, Vol. 32, No. 4, pp. 439-450, 1984.

[2] C. R. Paul, "Analysis of Multicondcutor Transmission Lines," New York: Wiley, 1994.

[3] C. P. Yuan and T. N. Trick, "A simple formula for the estimation of the capacitance of two-dimensional interconnects in VLSI circuits," IEEE Electron Device Letters, Vol. 3, No. 12, pp. 391-393, 1982.

[4] Y. W. Liu, K. Lan, and K. K. Mei, "Computation of capacitance matrix for integrated circuit interconnects using on-surface MEI method," IEEE Microwave and Guided Wave Letters, Vol. 9, No. 8, pp. 303-304, 1999.

[5] W. K. Sun, W. W. Dai, and W. Hong, "Fast parameter extraction of general interconnects using geometry independent measured equation of invariance," IEEE Transactions on Microwave Theory

and Techniques, Vol. 45, No. 5, pp. 827 836, 1997.
[6] R. Jin, Y. Cao, and Z. F. Li, "Fast parameter extraction for multicondcutor interconnects in multilayered dielectric media using mixture method of equivalent source and measured equation of invariance," IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part B, Vol. 20, No. 3, pp. 235-240, 1997
[7] H. Ymeri, B. Nauwelaers, and K. Maex, "Efficient procedure for capacitance matrix calculation of multilayer VLSI interconnects using quasi-static analysis and Fourier series approach," Journal of Telecommunications and Information Technology, pp. 40-44, 2002.
[8] J. Bernal, F. Medina, and M. Horno, "Quick quasi-TEM analysis of multiconductor transmission lines with rectangular cross section," IEEE Transactions on Microwave Theory and Techniques, Vol. 45, No. 9, pp. 1619-1626, 1997.
[9] H. H. Park, J. H. Kwon, J. W. Lee, and H. J. Eom, "Analysis of single-layered multiconductor transmission lines using the Fourier transform and mode-matching techniques," Microwave and Optical Technology Letters, Vol. 36, No. 4, pp. 315-317, 2003.
[10] M. Kamon, N. A. Marques, L. M Silveria, and J. White, "Automatic generation of accurate circuit models of 3-D interconnect," IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part B, Vol. 21, No. 3, pp. 225240, 1998.
[11] A. Papachristoforos, "Method of lines for analysis of planar conductors with finite thickness," IEE Proc. Microw. Antennas Propag. 141, pp. 223-228, 1994.
[12] J. T. Kuo, "Accurate quasi-TEM spectral domain analysis of single and multiple coupled microstrip lines of arbitrary metallization thickness," IEEE Transactions on Microwave Theory and Techniques, Vol. 43, No. 8, pp. 1881-1888, 1995.
[13] V. Veremey and R. Mittra, "A technique for fast calculation of capacitance matrices of interconnect structures," IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part B, Vol. 21, No. 3, pp. 241-249, 1998.
[14] F. Xiao, M. Norgen, and S. He, "Quasi-TEM approach of coupled-microstrip lines and its application to the analysis of microstrip filters," International Journal of RF and Microwave Computer-Aided Engineering, Vol 22, No. 1, pp. 131-139, 2012.
[15] Y. I. Bontzios, M. G. Dimopoulos, and A. A. Hatzopoulos, "An evolutionary method for efficient computation of mutual capacitance for VLSI circuits based on the method of images,"Simualtion Modelling Practice and Theory, Vol. 19, pp. 638-648, 2011.
[16] Y. B. Park and G. H. Park, "Capacitance of a conductor-backed coplanar waveguide with an upper shielding," Microwave and Optical Technology Letters, Vol. 53, No. 6, pp.1364-1368, 2011.
[17] T. Demeester and D. De Zutter, "Eigenmode-based capacitance calculations with applications in passivation layer design," IEEE Transactions on Components Packaging and Manufacturing Technology, Vol. 1. No. 6, pp. 912-919, 2011.
[18] P. N. Harms, C. H. Chan, and R. Mittra, "Modeling of planar transmission line structures for digital circuit applications," Arch. Eleck. Ubertragung., Vol. 43, pp. 245-250, 1989.
[19] Y. R. Crutzen, G. Molinari, and G. Rubinacci (eds.), "Industrial application of electromagnetic computer codes," Norwell, MA: Kluwer Academic Publishers, p. 5, 1990.

ISSN 2210-142X

A Multi-view 3D Shape Reconstruction System using Level Sets

Moumen T. El-Melegy1 and Nagi H. Al-Ashwal2
1Electrical Engineering Department, Assiut University, Assiut 71516, Egypt
moumen@aun.edu.eg
2Electrical Engineering Department, Ibb University, Ibb, Yemen
nlashwal@yahoo.com

ABSTRACT

The 3D reconstruction of a scene from multiple images is a fundamental problem in the field of computer vision. Existing methods can be classified into two strategies: bottom-up or top-down. This paper presents a full system for complete 3D shape reconstruction following the top-down strategy. A rotary table is employed to change a camera's viewing direction to an object on the table. This offers a cost-effective solution to the multi-view stereo acquisition problem without the need for using several cameras. From the acquired calibrated images of the object, a variational approach is developed for 3D shape reconstruction of the object. The approach works directly in 3D Euclidean space based on a level set formulation. A correlation criterion between the 2D images is optimized by driving the evolution of the surface using the corresponding Euler-Lagrange equation. Several successful experiments to evaluate the proposed system are reported.

Keywords: Level set methods, Multi-view stereo and 3D reconstruction.

1. INTRODUCTION

The 3D reconstruction of a scene from multiple images is a fundamental problem in the field of computer vision. There are many real world applications of 3D models, such as computer graphics, robot navigation, TV/film special effects, computer games, virtual reality inspection, navigation, and object identification. Recently it has become a very important and fundamental step in particular for cultural heritage digital archiving. The motivations are different: documentation in case of loss or damage, virtual tourism and museum, education resources, interaction without risk of damage, and so forth. often be close together (i.e., small baseline) so that correspondence techniques are effective.

Correspondences must be maintained over many views spanning large changes in viewpoint. The modeling of occlusions is complicated and there is not a standardized and widely accepted framework for modeling occlusions for bottom-up methods [4][5]. There are many methods in literature belonging to this strategy. Refer to [1][4][27]for a recent review and taxonomy of algorithms. Given a set of images of a scene captured from multiple calibrated cameras, the goal is to recover the unknown 3D structure using these images and the knowledge of the camera geometry. This problem is known as multi-view stereo in computer vision and is of great importance as it provides a way to infer the geometric and photometric properties of a scene without interfering with it. Additional advantages of using multi-view stereo to infer these scene properties include that the setup is fairly simple, the cost is relatively low and the process can be easily automated.

In most applications in the field of vision-based 3D reconstruction [1][4], two strategies can be applied:

• Bottom-up or data driven strategies [4][5]: These algorithms are based on image matching, using either intensity-based (direct) methods or feature-based methods. This class includes all techniques that compute correspondences across images and then recover 3D structure by triangulation and surface

fitting. Establishing correspondence in this traditional way is a hard problem that often leads to many outliers ⟦1]. Some disadvantages of this approach are that, it is computationally intensive and algorithmically demanding. Furthermore, views must often be close together (i.e., small baseline) so that correspondence techniques are effective.
Correspondences must be maintained over many views spanning large changes in viewpoint. The modeling of occlusions is complicated and there is not a standardized and widely accepted framework for modeling occlusions for bottom-up methods ⟦4][5]. There are many methods in literature belonging to this strategy. Refer to⟦1][4][27]for a recent review and taxonomy of algorithms.

• Top-down or model driven strategies ⟦21]: In this case, the computations are performed in three-dimensional scene space in order to construct the volumes or surfaces in the world that are consistent with the input images. Top-down approaches assume there is a known, bounded area in which the objects of interest lie. These strategies overcome the disadvantages of the bottom-up approach; they have the ability to explicitly model occlusions and consider multiple views. Example shere include shape form silhouettes [12][13], space carving ⟦6][15][26], reconstruction using variational methods ⟦8][17][18][28][29], volumetric graph-cuts [30] and continuous global optimization [6]. The existing approaches suffer from one or more of the following issues in the problem of multi-view 3D reconstruction: shape representation, objective function optimization and the initialization requirements. In this paper we present a complete approach addressing these issues following the top-down strategy due to its notable advantages over the bottom-up strategies.
Many existing systems ⟦6][15][26][30] represent a shape as a set of voxels or polygon meshes, which may fail with complex shapes. In contrast, our proposed approach uses a level set representation that provides several advantages over traditional object representations, such as its capability to model complex surfaces and to cope with varying object topologies ⟦3]. In terms of optimization, several existing approaches iteratively update the shape based on local decisions starting from an initial volume, such as space carving methods ⟦6][15][26], which often causes the propagation of unrecoverable errors from one region to another. In contrast, our approach defines a global objective function on the whole 3D space, whose optimization leads naturally to the evolution to a partial differential equation (PDE) of the level set function. This evolution is efficiently solved on a discretized 3D grid using well-defined numerical methods ⟦20][23]. While many algorithms [12][13][8][17][18][28][29] require a good initialization to guarantee algorithm convergence, our approach tends to work starting from any initial position, and the reconstruction process is not sensitive to the initial level set function.
Our approach is inspired by Faugeras and Keriven ⟦8], who were the first to combine image matching and shape reconstruction in a variational framework that minimizes an energy functional that is written as the integral of a data fidelity criterion on the unknown surface. However, the numerical implementation is rather complicated and requires simplification by dropping some terms. The final result tends to be sensitive to where the level set function has started its evolution. In contrast, we start with a new, different formulation that models the surface to be reconstructed as a level set embedded in an energy functional from the beginning. An advantage of this formulation is that one can easily and in straightforward manner model any available information on the object shape into the energy functional. For example, this may allow the reconstruction of an object with shape variations consistent with a set of training model examples [24]. This can prove very useful indeed in several nowadays applications of computer vision and graphics that focus on building 3D models of a certain category of objects.
For example, generation of realistic 3D human face models and facial animations can indeed exploit the earlier knowledge that the object looks like a human face. Another example is the 3D modeling of the human jaw from a sequence of intra oral images [25], which can make use of prior information on the

shape of human teeth.

In addition to the distinctive and novel aspects of our approach (strong, flexible shape representation and efficient global optimization algorithm with no special initialization requirement), we build a simple, yet effective 3D reconstruction system consisting of a rotary-table and a USB camera, both controlled via a desktop PC. The system uses the rotary table to change the camera's viewing direction to an object on the table. This offers a cost-effective solution to the multi-view stereo acquisition problem without the need for using several cameras. A checkerboard calibration pattern is used to calibrate the camera in the very first view to recover the camera projective geometry and its parameters. Then the camera parameters are automatically updated for all the other views, without the need for re-calibration. The images acquired from the different views are used for shape modeling.

The developed approach is successfully evaluated in several experiments using synthetic and real datasets, as well as using our own system setup. The rest of this paper is organized as follows. In Section II we describe our acquisition setup for capturing images from multiple views. A brief description of camera geometry and calibration is also given. We explain our approach for 3D reconstruction in Section III. The experimental results are reported in Section IV. This paper is concluded in Section V.

II. SYSTEM SETUP AND CAMERA CALIBRATION

We have developed a data acquisition setup to capture calibrated images from multiple views of objects. The hardware setup consists of a rotary table, a USB camera and a desktop PC. The rotary table is built from scratch using a stepper motor, a driver and an interface circuit to the parallel port of the PC. Figure 1 shows the constructed rotary table. The system operates as follows: The object to be reconstructed is placed on the rotary table. In order to obtain different viewpoints of the object, we simply rotate the table by a desired angle each time and grab an image. This is repeated for a number of times (typically 10-12) to cover the object from all views. A complete program with graphical user interface (GUI), written in Visual C++, is used to control the speed, direction and the step size of the table motion and to acquire the images from the USB camera. By construction, the step size of the table rotation can be as low as 0.1 .

Most 3D reconstruction techniques require the calibration of the camera, especially if quantitative measurements are sought. Camera calibration allows us to derive the projection equations linking points in our 3D world to their projections on the image and solve for the camera intrinsic and extrinsic parameters. According to the most common camera model in computer vision, the pinhole camera model [9], a world 3D point $\tilde{\mathbf{X}} = (X, Y, Z, 1)^T$ and the corresponding image point $\tilde{\mathbf{x}} = (u, v, w)^T$ (the over-symbol ~ denotes homogeneous vectors) are related via

$$\tilde{\mathbf{x}} = \mathbf{P}\,\tilde{\mathbf{X}}, \qquad (1)$$

where **P is a 3 × 4** homogeneous camera projection matrix which can be decomposed into [9]:

$$\mathbf{P} = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right], \qquad (2)$$

where K is the camera calibration matrix containing the intrinsic parameters of the camera (e.g., focal length and principal point), R is an orthonormal rotation matrix and t is a 3D translation vector. Both R and t represent the camera extrinsic parameters which define the camera orientation and position with respect to the world coordinates. The above formulation relies on the assumption that the map from the world to the image is linear projective. That is, there exists no significant lens distortion or it has been corrected [22]. The procedure of calculating the camera matrix P is called camera calibration [9]. Many methods are available in literature [9] for the determination of the camera matrix. We use for this sake in our system Robert's technique [10], which has the advantage of high accuracy without the need of

accurate feature extraction. This is done with the help of a checkerboard pattern, see Figure 2. The world coordinate system is chosen along the sides of the calibration pattern. At the initial viewpoint (first position of the table), the camera is initially calibrated by putting the calibration pattern on the table such that the world's Y-axis coincidences with the rotation axis of the table (A correction procedure is carried out here to compensate for any possible misalignment [2]). This way the initial projection matrix 0 P calibrated. at this first viewpoint is

At each subsequent rotation angle, we need to compute a new projection matrix for the new viewpoint. Recalibrating the camera using the calibration pattern is not needed. By rotating the object on the table by a specific angle , the camera intrinsic parameters are expected not to change, but we simply need to update the extrinsic parameters of the camera to reflect the new camera orientation.

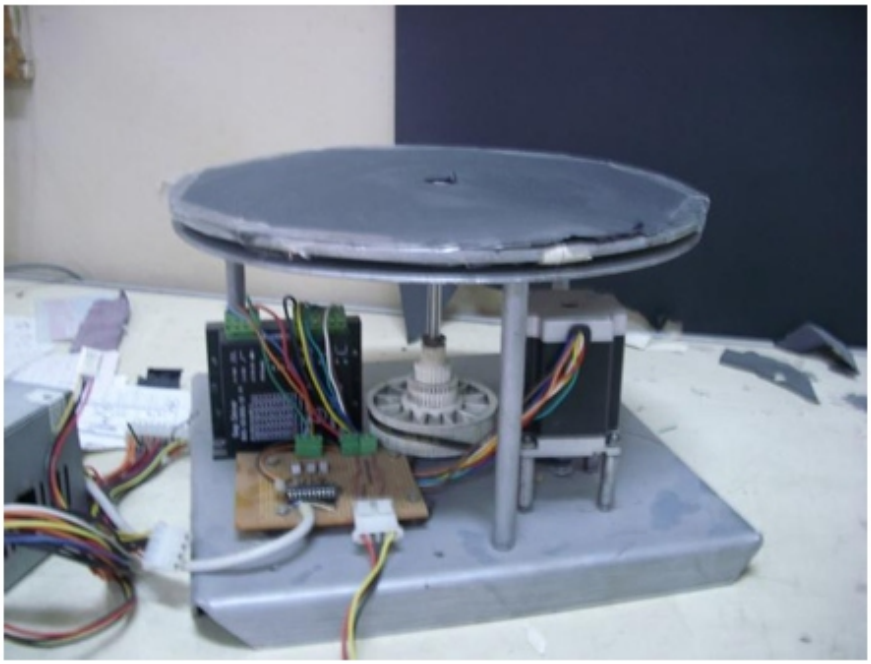


Figure 1. The rotary table is used in our work to change the camera's viewing direction to an object.

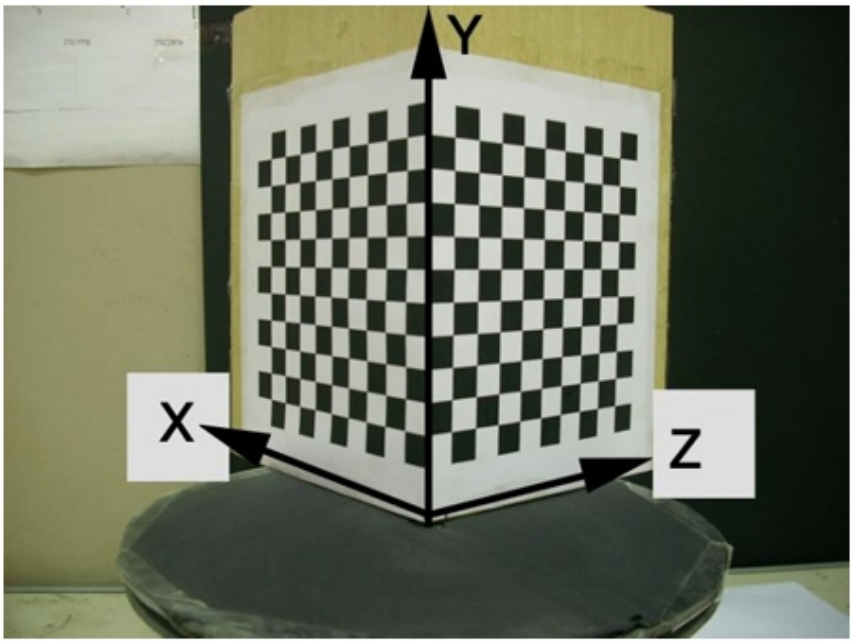


Figure 2. A checkerboard pattern is placed on the rotary table to calibrate the camera. The world's Y-axis coincidences with the table rotation axis. The camera projection matrix at any view i is calculated as

$$\mathbf{P}_i = \mathbf{P}_0 \mathbf{R}(\theta_i), \qquad (3)$$

where R (θ) I is a 4× 4 rotation matrix around the Y-axis by the angle θ

A simple experiment was carried out to assess the accuracy of camera calibration across multiple-views. Several images of the pattern from various views are used to reconstruct the corners of the pattern squares in the 3D world coordinates. These 3D points are then compared to the known ground-truth positions of the calibration pattern squares. The root-meansquare error is found to be about 0.1±0.04 mm across the various views, which indicates very high calibration accuracy.

III. PROPOSED APPROACH

In this paper we consider the problem of recovering the 3D shape of a given object from a set of 2D images taken from n multiple calibrated viewpoints. We assume that the object is made of Lambertian materials and there is texture in the albedo. The approach proposed in this work shares some fundamental points with that of [8]. The two approaches find a surface that minimizes an energy functional that is written as the integral of a data fidelity criterion on the unknown surface. This criterion is based on the normalized cross-correlation between image pairs. However ours is different in a number of aspects. In [8] the normalized cross-correlation between image pairs is done by picking a window around a point in one image and comparing it with its transformed window (around the corresponding point) in another image. This transformation is computed through the unknown surface, which is then taken locally planar. Here, we will present a better matching process. Instead of considering the transformations between image patches, we project a point on the surface to each visible image and then pick a window around each projected point. Then the matching is done by comparing each pair of these widows.

The Euler-Lagrange equation that minimizes the functional of [8] is driven in terms of the surface, then the evolution is implemented via introducing later a level set formulation with some simplification by dropping some terms. In this work, the energy functional is embedded in a level-set framework from the beginning. An advantage of this formulation is that one can easily and in straightforward manner model any available information on the object shape into the energy functional.

The minimization of the energy functional is done on the whole 3D space, leading naturally to the evolution to a partial differential equation (PDE) of the level set function. All the terms in the PDE are used to implement the evolution without dropping any of them. Consequently, our approach offers a number advantages over that of [8] and similar works, e.g., [17][18][29], which tend to be sensitive to the initial position of the surface. On the contrary, our approach tends to work starting from any initial position, and the final result is not sensitive to where the level-set function starts its evolution.

In this section, we give the full details of the proposed approach. We start with brief review of some fundamental and preliminary concepts on level sets. Then we formulate our approach in a level-set framework, addressing the object visibility and evolution issues.

A. Preliminary Concepts

Level set methods were devised by Osher and Sethian [3] to implicitly model evolving interfaces in two or three dimensions. Level set methods have several advantages compared to the explicit active contours (snakes) introduced by Kass et al. [11] and other deformable methods that use the parametric representation of curves and surfaces:

• One can perform numerical computations involving curves and surfaces on a fixed Cartesian grid without having to parameterize these objects.
• It is conceptually straightforward to move from two to three and even higher dimensions (although computational cost is exponential in dimension).
• Moving interfaces automatically handle the topological changes, which happen often and are desired in evolutions; they can easily merge or separate.
• Geometric quantities are easy to calculate: surface normal, curvature, direction and distance to the nearest point on the surface. Surface motion can depend on them.

Therefore, level set methods have received attention in many fields, including image processing, computer graphics, computer vision, fluid mechanics, and computational geometry. By now, level set methods have become standard tools for implementing evolution PDEs.

In an implicit formulation the interface in Ω n R is a non empty subset defined by [3]

$$\Gamma = \{\mathbf{x} \in \Omega; \phi(\mathbf{x}) = 0\}, \qquad (4)$$

where the function, $\phi(x)$, $\phi : R^n \to R$, has the following properties

$$\begin{cases} \phi(\mathbf{x}) > 0 & for \quad \mathbf{x} \in \text{inside}(\Gamma), \\ \phi(\mathbf{x}) < 0 & for \quad \mathbf{x} \in \text{outside}(\Gamma), \\ \phi(\mathbf{x}) = 0 & for \quad \mathbf{x} \in \Gamma. \end{cases} \qquad (5)$$

In three spatial dimensions the gradient of the implicit function $\phi(\mathbf{x})$, is given by $\nabla\phi = \left(\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial z}\right)$. $\nabla\phi$ perpendicular to the isocontours of Φ pointing in the direction of increasing Φ. Thus, the unit normal is $\boldsymbol{n} = \frac{\nabla\phi}{|\nabla\phi|}$, for points on the interface. The mean curvature of the interface is defined as the divergence of the normal, $\kappa = \nabla.\boldsymbol{n} = \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$, that $k > 0$ for convex regions, $k < 0$ for concave regions and $k = 0$ for a plane.

Often, the interior of the interface (surface) can be presented as $\{\mathbf{x} \in \Omega : H(\phi(\mathbf{x})) = 1\}$, where $H(.)$ standard Heaviside function. Using this notation, the integration of some function f over the interior can be given by $\int_\Omega f\, H(\phi(\mathbf{x}))\, d\mathbf{x}$. The directional derivative of the Heaviside function H in the normal direction n is given by $\nabla H(\phi(\mathbf{x})).\boldsymbol{n} = \delta(\phi)|\nabla\phi|$, where $\delta(.)$ is the standard Dirac function on the real line $(\delta(u) = \partial H / \partial u)$. As such, the integral of the function f over only the boundary Γ can be presented by $\int_\Omega f\, \delta(\phi(\mathbf{x}))|\nabla\phi(\mathbf{x})|\, d\mathbf{x}$.

B. Multi-view Stereo Level-set Formulation

The object to be reconstructed is to be represented by a level set function Φ. We seek the zero level set of Φ that minimizes the energy functional $E(\phi) = \int_\Omega \Phi(\mathbf{X})\ \delta(\phi)|\nabla\phi|\ dXdYdZ +$

$$\mu \int_\Omega \delta\ (\phi)|\nabla\phi|\ dXdYdZ\,, \quad (6)$$

where $\Phi(X)$ is some matching score for all points $\mathbf{X} = (X, Y, Z) \in \Omega \subset R^3$, which should be minimized on only the object surface (boundary). The integral (6) is however done on the whole 3D space Ω by introducing the term $\delta(\phi)|\nabla\phi|$ in the integral. The second term is a smoothness term weighted in the functional by the regularizing parameter *u* that to be chosen a priori. One advantage of this formulation is that one can easily model any available information on the object shape into the energy functional (6);

for example we can add a priori information to force the level set function to reconstruct a predefined object.

The function $\Phi(X)$ is based on the normalized cross correlation and taken as

$$\mathbf{\Phi}(\mathbf{X}) = \sum_{\substack{i,j=1 \\ i \neq j}}^{n_v} \mathbf{\Phi}_{ij}(\mathbf{X}) \qquad (7)$$

where v n is the number of visible cameras to the current voxel X and the cross-correlation, Φij (X) , between two visible cameras iand j is given by[18]

$$\mathbf{\Phi}_{ij}(\mathbf{X}) = 1 - \frac{< I_i, I_j >}{\sqrt{< I_i, I_i > \cdot < I_j, I_j >}}. \qquad (8)$$

where the inhomogenizing transformation π (converts form homogenous to inhomogeneous coordinates). The quantities in (8) are thus given by

$$(r_i, c_i) = \boldsymbol{\pi}(\tilde{\mathbf{x}}_i) = \boldsymbol{\pi}(\mathbf{P}_i\tilde{\mathbf{X}}) = \boldsymbol{\pi}(u_i, v_i, w_i) = \left(\frac{u_i}{w_i}, \frac{v_i}{w_i}\right), \qquad (9)$$

$$< I_i, I_i >= \sum_{m=-\frac{h}{2}}^{\frac{h}{2}} \left(\sum_{n=-\frac{w}{2}}^{\frac{w}{2}} \left(I_i\ r_i + m, c_i + n\right) - \overline{I}_i\right)^2, \qquad (10)$$

$$< I_j, I_j >= \sum_{m=-\frac{h}{2}}^{\frac{h}{2}} \left(\sum_{n=-\frac{w}{2}}^{\frac{w}{2}} \left(I_j\ r_j + m, c_j + n\right) - \overline{I}_j\right)^2, \qquad (11)$$

$$< I_i, I_j >= \sum_{m=-\frac{h}{2}}^{\frac{h}{2}} \sum_{n=-\frac{w}{2}}^{\frac{w}{2}} \left(I_i\left(r_i + m\ c_i + n\right) - \overline{I}_i\right) \times \left(I_j(r_j + m, c_j + n) - \overline{I}_j\right), \qquad (12)$$

where h and w are the height and width of the correlation window. We take h=5, and w=5 in all our experiments. The quantities i I and j I denote the mean values of i I and j I respectively

$$\overline{I}_i = \frac{1}{h \times w} \sum_{m=-\frac{h}{2}}^{\frac{h}{h}} , \sum_{n=-\frac{w}{2}}^{\frac{w}{2}} I_i \left(r_i + m \; c_i + n \right), \qquad (13)$$

$$\overline{I}_j = \frac{1}{h \times w} \sum_{m=-\frac{h}{2}}^{\frac{h}{h}} , \sum_{n=-\frac{w}{2}}^{\frac{w}{2}} I_j \left(r_j + m \; c_j + n \right). \qquad (14)$$

Summation that appears in (7) is computed only for those points of the surface S = { X:Φ (X) =0} which are visible in the two concerned images. Thus, estimating Φ(X) requires computing the hidden parts of the surface for all cameras.

C. Visibility

At each step, in order to compute the summation in (7), we need to compute the visibility, *X*. It is equivalent to the problem of determining which part of the surface is visible from a given view point (the camera center in our case). This is a classical problem in computer graphics. A typical approach to this problem is the so-called ray tracing. The idea is to start from each point in the domain of interest, shoot a ray towards the view point, and check the number of times this ray hits the surface. Unfortunately this intuitive algorithm turns out to be computationally expensive. However, it is possible to exploit the level-set representation of surfaces to efficiently solve the problem. We adopt here a level-set implementation of the implicit ray tracing technique that is originally reported in [19]. This is a one-pass algorithm that finds the line of sight for a given configuration of implicit surfaces in an incremental way. The algorithm computes another level set function , which tells us the portions of Φ that are visible from the view point. More precisely, $\{\mathbf{X} \in \Omega : \psi(\mathbf{X}) \geq 0\}$ will be the regions visible from a view point v, see Figure 3. Therefore, the desired visibility function can be written as $\chi = H(\psi)$. For more detail on implicit ray tracing, we redirect readers to [19].

D. Evolution Equation

Now we can rewrite (7) to reflect the visibility as:

$$\mathbf{\Phi} = \sum_{\substack{i,j=1 \\ i \neq j}}^{n} (\chi \; i, \mathbf{X}) \chi(j, \mathbf{X}) \mathbf{\Phi}_{ij}, \qquad (15)$$

where n is the number of all views, X (z X ,) is a characteristic function which denotes the visibility of the voxel X to the camera z

$$\begin{cases} \chi(z, \boldsymbol{X}) = 1 & \text{if } \mathbf{X} \text{ is visible to camera z,} \\ \chi(z, \boldsymbol{X}) = 0 & \text{if } \mathbf{X} \text{ is not visible to camera z.} \end{cases} \qquad (16)$$

The Euler-Lagrange equation for Φ of the functional (6) can be shown (after some lengthy computation) equal to

$$\frac{\partial E}{\partial \phi} = -\delta(\phi)\left(\nabla \mathbf{\Phi} \cdot \boldsymbol{n} + \mathbf{\Phi}\kappa + \mu\kappa\right), \qquad (17)$$

where k is the mean curvature. Employing an artificial time $t < 0$, the evolution equation becomes

$$\frac{\partial \phi}{\partial t} = -\frac{\partial E}{\partial \phi} = \delta(\phi)\left(\nabla \mathbf{\Phi} \cdot \boldsymbol{n} + \mathbf{\Phi}\kappa + \mu\kappa\right). \qquad (18)$$

Note here, on the contrary to other approaches [8][18] we neither drop any terms nor approximate the other terms; all the terms of the resulting evolution equation are used.
In our implementation, we use a regularized form of $\delta(\phi)$ [20]

$$\delta_\varepsilon(\phi) = \frac{1}{\pi}\,\frac{\varepsilon}{\varepsilon^2 + \phi^2}\ . \qquad (19)$$

This regularized form $\delta_\varepsilon(\phi)$ is used in (18) in place of $\delta(\phi)$. Using this approximation, the algorithm has the tendency to compute a global minimizer. One of the reasons is that, the Euler-Lagrange equation acts only locally, on a few level surfaces around Φ = 0 using the original Dirac function, while by the regularized form, the equation acts on all level sets, of course stronger on the zero level set, but not only locally. In this way, in practice, we can obtain a global minimizer, independently of the position of the initial set. The final evolution equation is hence given by

$$\frac{\partial \phi}{\partial t} = -\frac{\partial E}{\partial \phi} = \delta_\varepsilon(\phi)\left(\nabla \mathbf{\Phi} \cdot \boldsymbol{n} + \mathbf{\Phi}\kappa + \mu\kappa\right). \qquad (20)$$

We now turn to some implementation issues. The term $\nabla \mathbf{\Phi} = \frac{\partial \mathbf{\Phi}}{\partial \mathbf{X}}$ is given by

$$\nabla \mathbf{\Phi} = \sum_{\substack{i,j=1 \\ i=j}}^{n} \chi(i,\mathbf{X})\chi(j,\mathbf{X})\nabla \mathbf{\Phi}_{ij}. \qquad (21)$$

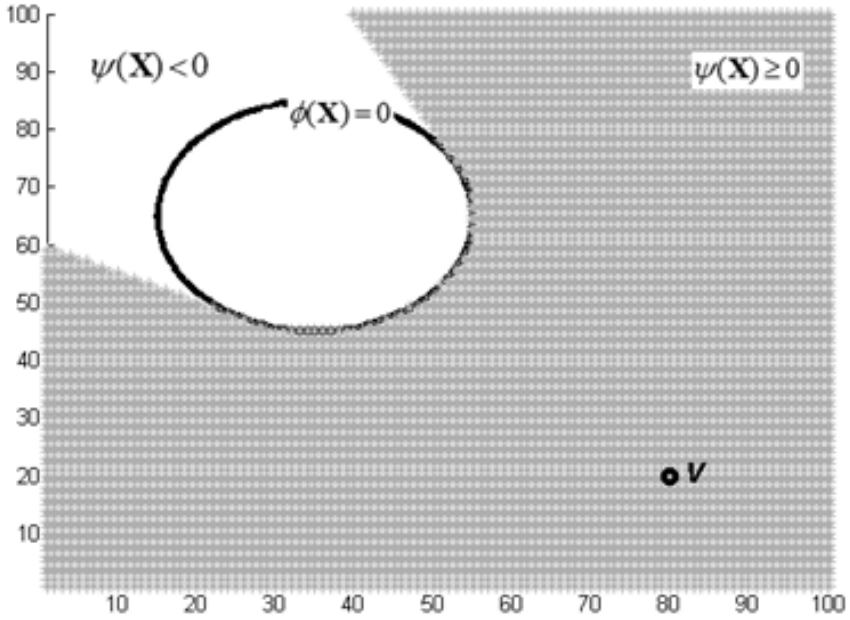


Figure 3. Illustration of visibility algorithm in 2D. The view point v is visible to all points in the region $\psi(\mathbf{X}) < 0$ the gray region, and invisible to the region $\psi(\mathbf{X}) \geq 0$, the white region.

The gradient $\nabla \mathbf{\Phi}_{ij}$ can be computed directly from (8). However we will need to calculate the following derivatives: $\frac{\partial I_i}{\partial \mathbf{X}}$ and $\frac{\partial I_j}{\partial \mathbf{X}}$, which can proceed as follows. Using the chain rule we have

$$\frac{\partial}{\partial \mathbf{X}} I_i\left(\boldsymbol{\pi}\left(\mathbf{P}_i \tilde{\mathbf{X}}\right)\right) = \frac{\partial I_i(\mathbf{P}_i \tilde{\mathbf{X}})^T}{\partial \boldsymbol{\pi}} \cdot \frac{\partial \boldsymbol{\pi}\left(\tilde{\mathbf{x}}_i\right)}{\partial \tilde{\mathbf{x}}_i} \cdot \frac{\partial \tilde{\mathbf{x}}_i}{\partial \tilde{\mathbf{X}}} \cdot \frac{\partial \tilde{\mathbf{X}}}{\partial \mathbf{X}} \quad (22)$$

where

$$\frac{\partial I_i(\mathbf{P}_i \tilde{\mathbf{X}})^T}{\partial \boldsymbol{\pi}} = \nabla I_i, \quad (23)$$

and

$$\frac{\partial \boldsymbol{\pi}(\tilde{\mathbf{x}}_i)}{\partial \tilde{\mathbf{x}}_i} = \frac{\partial (u_i / w_i, v_i / w_i)}{\partial (u_i, v_i, w_i)} = \begin{bmatrix} 1/w_i & 0 & -u_i / w_i^2 \\ 0 & 1/w_i & -v_i / w_i^2 \end{bmatrix}. \quad (24)$$

And two last ingredients needed,

$$\frac{\partial \tilde{\mathbf{x}}_i}{\partial \tilde{\mathbf{X}}} = \mathbf{P}_i \quad \text{and} \quad \frac{\partial \tilde{\mathbf{X}}}{\partial \mathbf{X}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \quad (25)$$

As such, we have all pieces need to compute $\frac{\partial I_i}{\partial \mathbf{X}}$ and $\frac{\partial I_j}{\partial \mathbf{X}}$, which allows us to calculate $\nabla \mathbf{\Phi}_{ij}$ then $\nabla \mathbf{\Phi}$.

The numerical implementation of the PDE evolving (20) is carried out on the discretized Cartesian grid using the semiimplicit scheme [20][23] to satisfy the Courant-FriedrichsLewy (CFL) condition [23]. By using this scheme we can speed up the evolution of the level set function.

Implementation note: The background of the object can be segmented out from the input images beforehand. So the background pixels in each image can be given a specific color (e.g., black). A better scenario though is to mark those background pixels with alternating colors in the input images (e.g., white in one image and black in another, and so on). This will cause those points to have very little correlation among the various views (thus contributing significantly to the error criterion (6)). Consequently, this will softly guide the level set evolution to exclude from the object the 3D points being projected to background pixels in any of the input images. This scenario improves further the results of the approach in a straightforward manner, without the need for any modification in the level set evolution, and more importantly, without the need for taking hard decisions on those points, the case that may lead to unrecoverable reconstruction errors.

IV. EXPERIMENTAL RESULTS

In this section, the proposed approach is evaluated extensively using several experiments on different

datasets. Firstly the approach (implemented in Matlab, but the data acquisition program with the GUI implemented in Visual C++), is applied to a synthetic dataset generated using the AutoCAD program. This experiment using the AutoCAD environment helps quantify the performance versus ground-truth results under varying degrees of artificial noise. Then the practical usefulness of the proposed approach is demonstrated through its application to real datasets publically available from the internet, as well as real datasets obtained using the system setup which we have constructed.

A. Synthetic Dataset

We apply our approach to a synthetic dataset generated using the AutoCAD 2007 program. We use AutoCAD program to simulate the developed system. As described in section II, the camera must be calibrated at the initial view. In this experiment, we accomplish this using a synthetic calibration pattern, see Error! Reference source not found.4. The captured image is then used to find 0 P as explained in Section II, from which all the other projection matrices are derived given the rotation angle of the object (rotary table). Then the setup is used to get 12 images of a horse object by rotating the object about the Y axis and taking an image every 30º, see Figure 5. The proposed approach is applied to those images, and some rendered views of the reconstructed object are shown in Figure 6. Apart from some voxelization effect due to the numerical implementation on a discretized grid, the horse shape is accurately modeled. Due to our efficient shape representation, notice how the fine details of the horse's legs and tail are correctly reconstructed. The size of volume in this experiment is 160×160×160 voxels, and it takes about 65 minutes to complete on a P4 3GHz PC with 2Gbytes RAM. The initial level set is a sphere with radius 0.01.

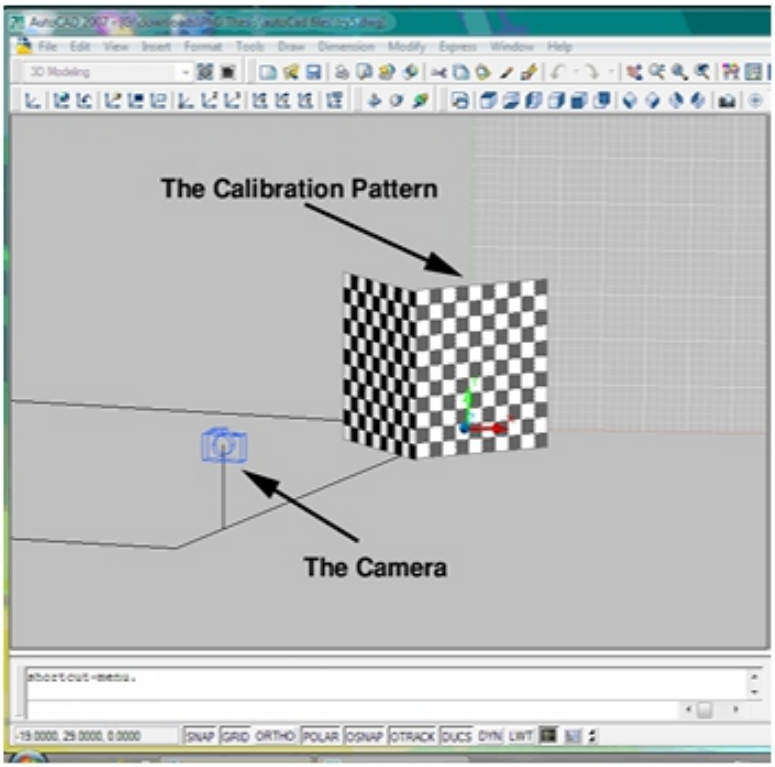


Figure 4. Using AutoCAD to simulate our developed setup and acquire the image of the calibration pattern.

To simulate non-ideal conditions in real environments, input images are noised by additive Gaussian

noise with zero mean and standard deviation, σ, that is varied from 0 to 70 in steps of 10. Then at each value of σ, our approach is applied to the input images. To assess the quality of the reconstructed shape, the 3D reconstruction from the noisy images is reprojected onto the different view directions and the silhouettes are obtained as illustrated in Figure 7. As an accuracy measure, we use the root mean square error (RMS E) between those obtained silhouettes and the corresponding ground-truth (noise-free) silhouettes:

$$E_{RMS} = \sqrt{\frac{1}{n_I H_I W_I} \sum_{i=1}^{n_I} (I_i^{proj} - I_i^{gt})^2} , \qquad (26)$$

where I n is the number of images, I H is the image height, I W is the image width, proj I I is the projected image to the i-th view, and gt I I is the corresponding ground-truth noise-free image for the i-th view. For the sake of comparison, the same procedure is repeated using Faugeras and Keriven's approach [8]. The plot of RMS E versus for the two approaches is shown in Figure 8. From this figure, one can notice consistently the better and robust performance of the proposed method over the approach in [8] versus all levels of noise. Up to the high noise level of 50, RMS E remains below 0.18 for our approach, while the other approach sooner exceeds this RMS level. For noise higher than that, the input images hardly show the details of the horse, and accordingly E RMS starts to increase notably. However, as shown in Figure 7(d) and (e), even in this case, the silhouettes of the reconstructed horse are recovered with rather good accuracy.

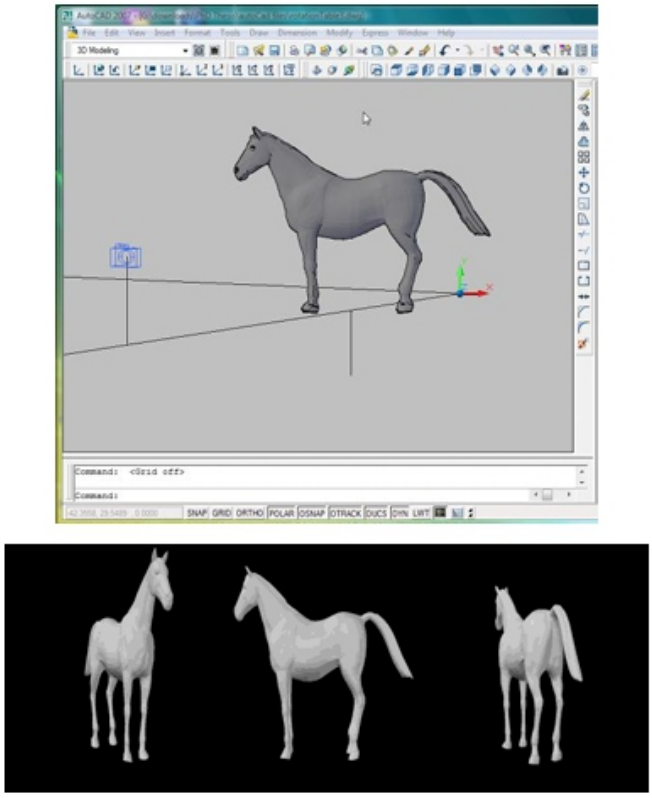


Figure 5. Using AutoCAD to simulate our developed setup and acquire images from multi-views of an object. (Top) Horse. (Bottom) Some views of 12-frame horse sequence taken using AutoCAD simulated camera.

Figure 6. Some rendered views of the final reconstructed 3D model by the proposed approach.

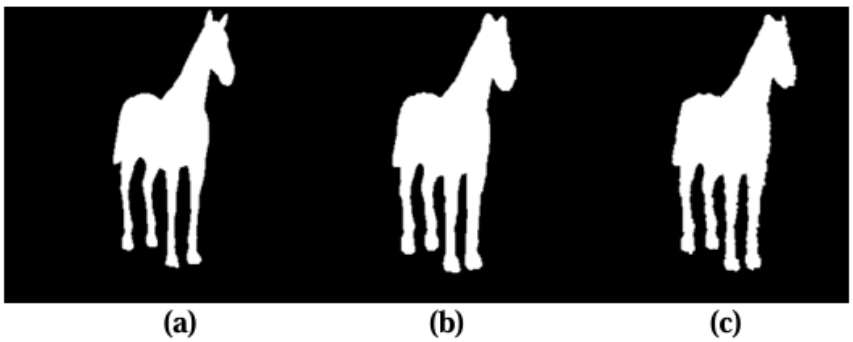
Note also that in our experiment with the approach in [8], care is taken to initialize it properly as it is sensitive to where the evolution has started, which is not the case with our approach.

B. Real Datasets from the Internet

To evaluate our approach on real objects we apply it first to reconstruct objects whose datasets are available on the internet. Each dataset contains the projection matrices of each view The backgrounds of the downloaded images are segmented manually.

Figure 9 shows some images of a 12-frame sequence for a rooster dataset obtained from the Computer Vision and Image Processing Lab at the University of Louisville [33]. To run our approach, the initial zero level set function was taken as a sphere, where several locations and sizes have been experimented with. Figure 10(a) illustrates some of those initial zero level sets along with the reconstructed 3D rooster model. The proposed approach was able to obtain very good reconstruction results starting from various initial level sets.

Notice also how the rooster's crown that has sharp and thin parts is accurately reconstructed. You can compare the results in Figure 10(a) with the results of the space carving approach in Figure 10(b), which showed a noisy result with missing parts and several floating voxels. Our results here are smoother and there are no floating voxels in the obtained 3D model. Some evolution stages are shown in Figure 11. In this experiment we used a volume of size 80×80×80, and =100. On a P4 with speed 2.8 GHz PC with 1GBytes of RAM; it takes about 30 minutes to reach the final shape.



(a) (b) (c)

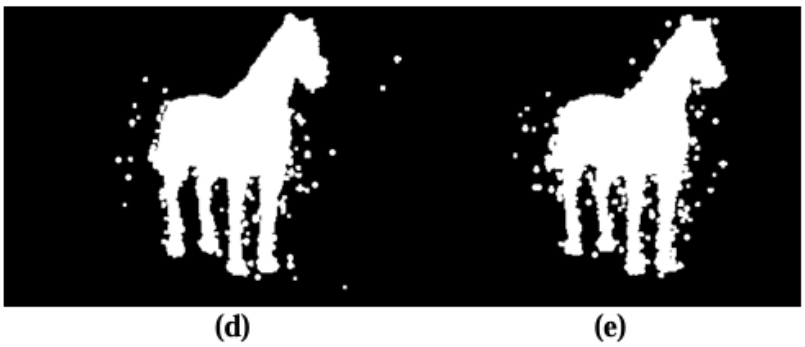


Figure 7. The ground-truth (noise-free) silhouette of the first view of the horse sequence in (a) versus the silhouettes of the projections of the reconstructed shape onto the first view at different values of σ : (b) σ =0, (c) σ =50, (d) σ =60, (e) σ =70.

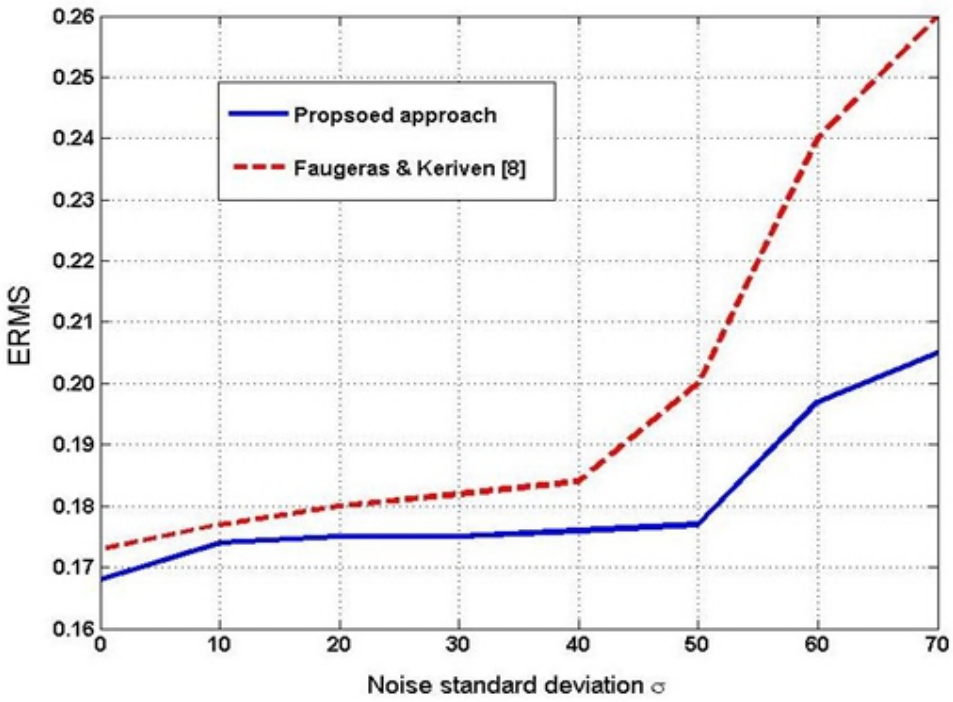


Figure 8. RMS E versus the noise standard deviation σ for the proposed approach and Faugeras and Keriven's approach [8] on the horse sequence.

Our method is also applied to the Oxford dinosaur dataset [31]. Figure 12 shows some of the 36 images used. Figure 13 shows the results of applying our method starting from different initial level set functions. As shown in this figure our approach could reconstruct the object regardless of the position and size of the initial level set function. The dinosaur's hands, feet and tail are correctly reconstructed in very good details. The volume size used in this experiment is 140×140×140 and u =100.

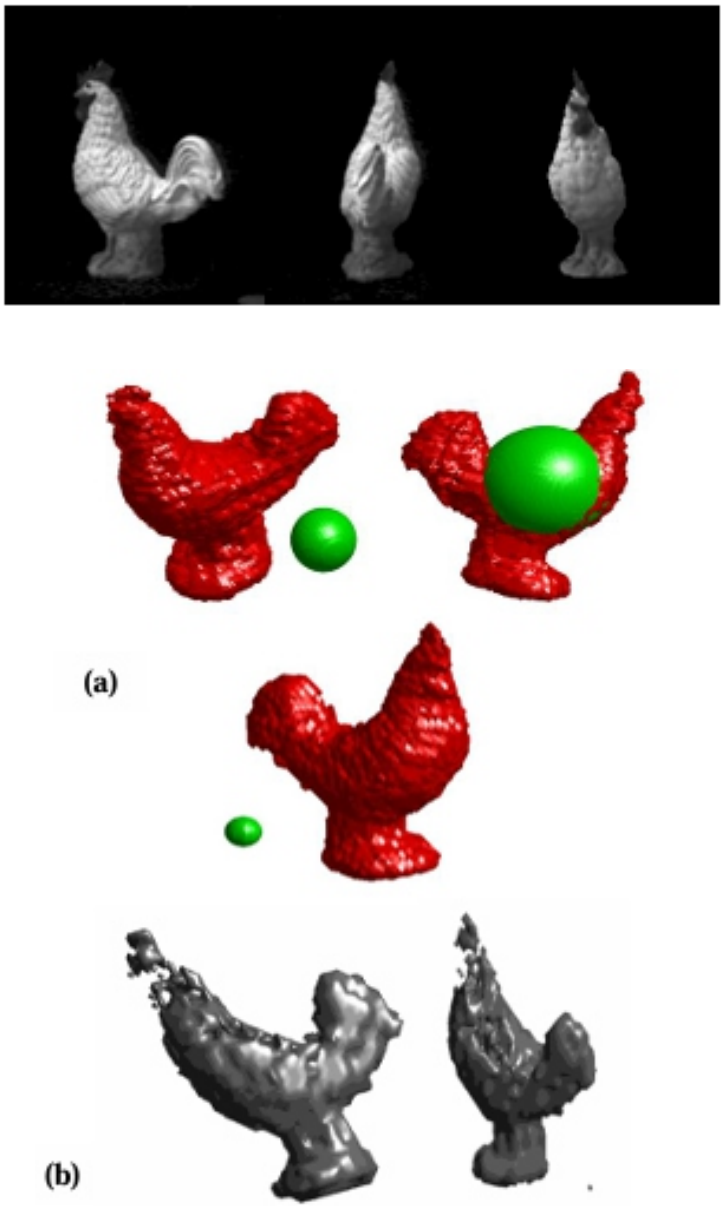


Figure 10. (a) Some initial level set functions (spheres shown in green) and the final reconstructed 3D model (shown in red) by the proposed approach. (b) Two rendered views of the reconstruction by the space carving technique [14] using the same input images.

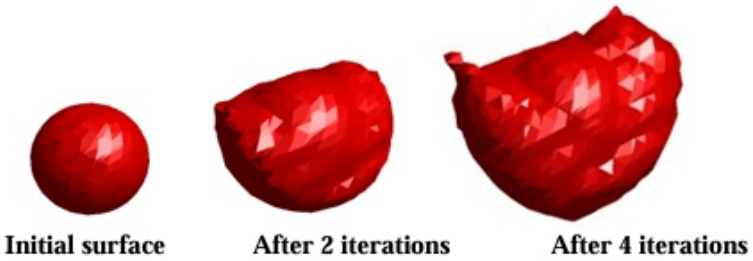


Figure 11. The evolution of the level set function to reconstruct the final surface for the experiment on the 12-frame rooster sequence.

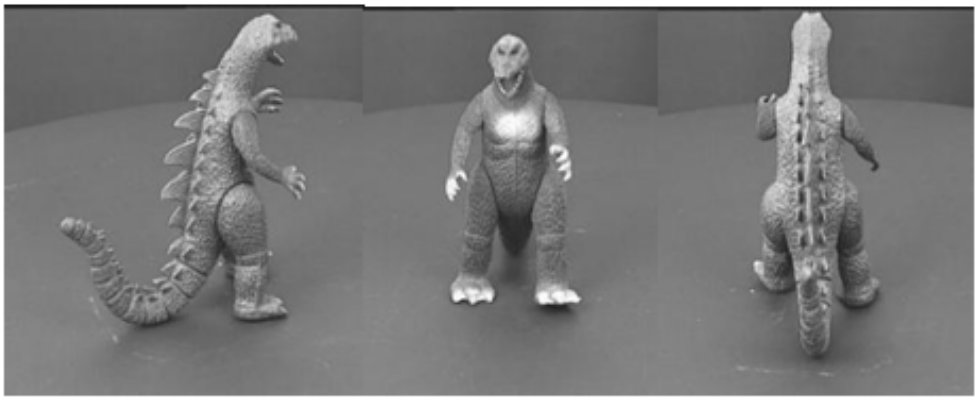


Figure 12. Some views of 36-frame dinosaur sequence.

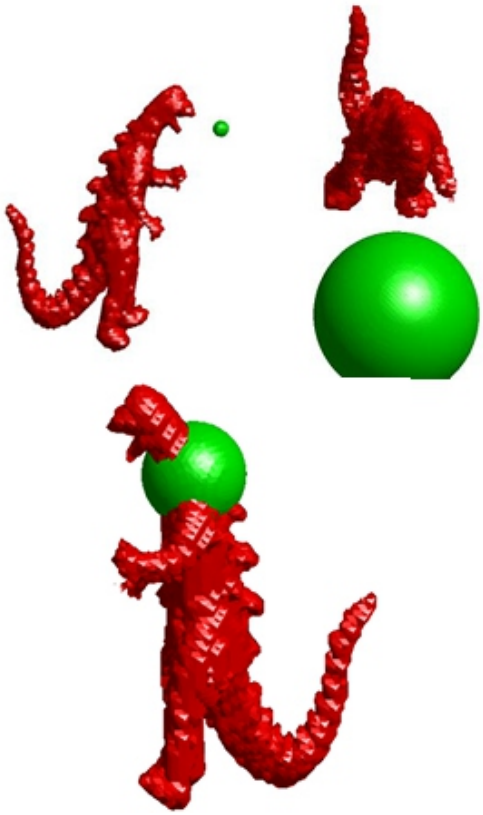


Figure 13. Several different initial level sets (spheres shown in green) and the final 3D surface (shown in red) for the dinosaur sequence. Our approach is also applied to a sequence of 24 images for another dinosaur [32]. Error! Reference source not found. shows some of these images. In Figure 14, we see the result of applying this method, as before, starting from different initializations. As shown in figure, the fine details of dinosaur's fingers and tail are accurately modeled. The volume size used in this experiment is 100×100×100 and u =100. It takes about 1 hour to get the final surface in 4 iterations.

C. Real Datasets using Our Setup

To evaluate the proposed approach using the developed setup, we apply it to some other objects. Figure 16 shows some of 12 images of a baby toy. The background in the input images has been segmented out manually. Figure 16 shows some views of the reconstructed object. In this experiment we used a volume of size 80×80×80, and u =100. The approach needed four iterations to reach the final shape taking about 30 minutes on a P4 with speed 2.8 GHz with 1GBytes of RAM. Another experiment is done on a duck toy for which, 12 images are taken by our setup, see Figure 17. Figure 18 shows some views of the reconstructed object. In this experiment we used a volume of size 80×80×80, and u =100. The four iterations needed to reach the final shape took about 20 minutes.

Figure 14. Some views of 24 images of dinosaur #2

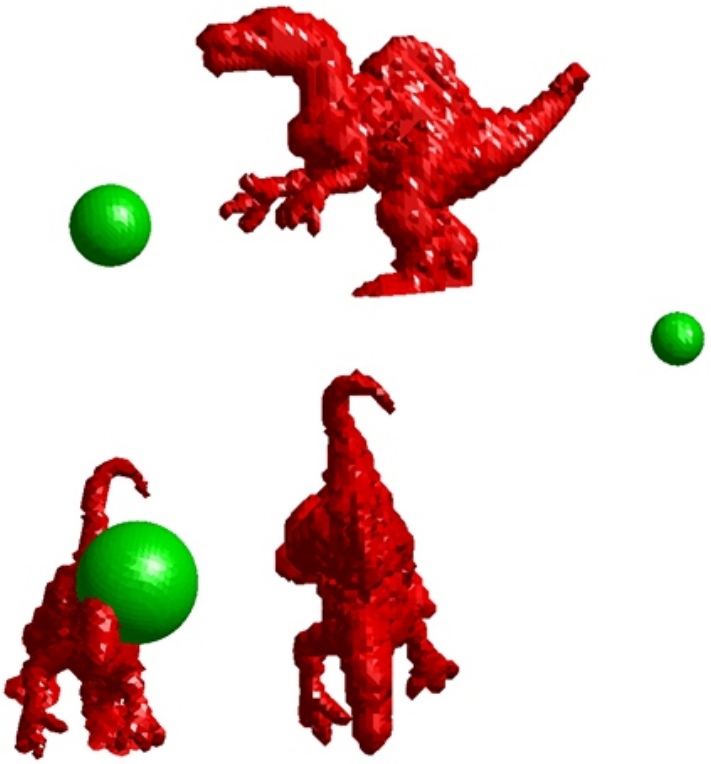


Figure 14. Several initial level set functions (spheres shown in green) and some views of the final reconstructed 3D model (shown in red) for dinosaur #2.

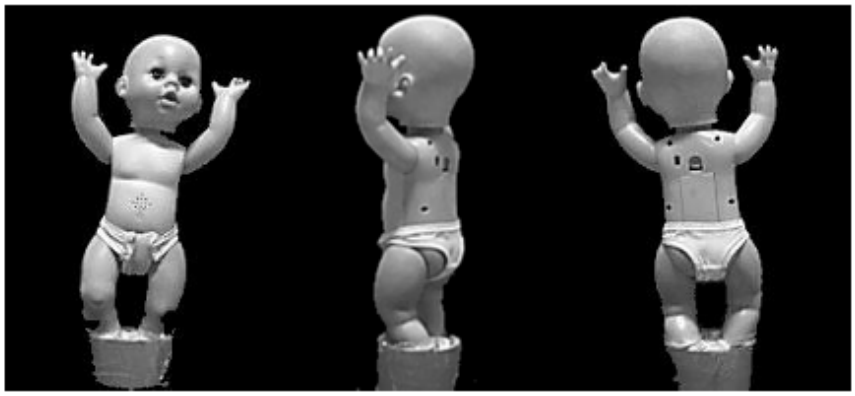


Figure 15. Some images of a 12-frame sequence of a baby toy taken by our setup.

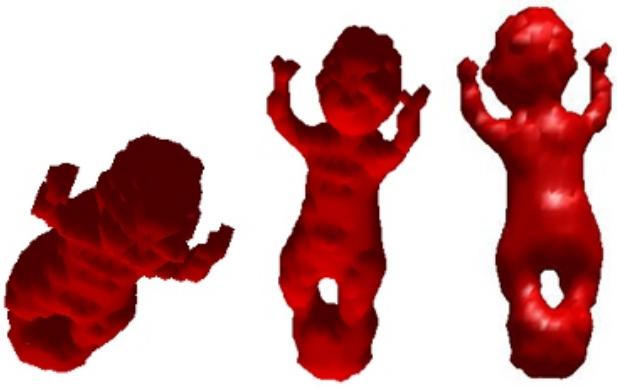


Figure 16. Some views of the final reconstructed baby toy.

Figure 17. Some images of a 12-frame sequence of a duck toy taken by our setup.

V. CONCLUSIONS

We have presented a simple, yet effective system for complete multi-view 3D shape reconstruction consisting of a rotarytable and a USB camera, both controlled via a desktop PC. The system offers a cost-effective solution to the multi-view stereo acquisition problem without the need for using several cameras. A variational approach has been formulated and developed to reconstruct the 3D object shape from the acquired sequence of calibrated images. In contrast to existing methods, this approach presents a flexible shape representation and an efficient optimization algorithm with no special initialization

requirement. The object is represented as a level set from the first problem formulation. This also allows the easy incorporation of any available shape a priori information in the energy functional in order to guide the surface evolution. Our extensive experimental results have shown the proposed approach can successfully find the 3D shape regardless of the position of the initial surface. Results have also demonstrated that fine details of the objects have been correctly recovered.

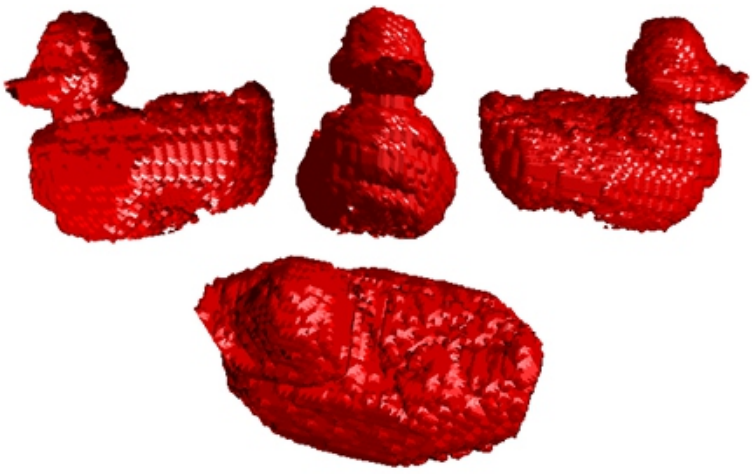


Figure 18. Some views of the final reconstructed duck toy.

Our current efforts are directed to utilize the ease of incorporating prior shape information in our approach in order to reconstruct objects with shape variations consistent with a set of training model examples. This can offer a great advantage when working with a specific category of objects. Some early implementation and results of our idea are already drafted in [24]. Another direction for our current research efforts is to address the time performance of the approach. The developed approach takes a small number of iterations (typically 4-8) to converge, but a single iteration may take rather a long time (about 3-12 minutes depending on the volume size) due to the high computational cost for the various approach operations (e.g., visibility calculation and level set evolution). One possibility to reduce this time is converting our shape reconstruction code from Matlab to a fully compiled programming language (e.g., C++). Another important possibility is to utilize parallel programming concepts to carry out computations concurrently on multi-core CPUs available nowadays on desktop Pcs.

REFERENCES

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", Int. J. of Computer Vision, 47(1-3), pp. 7–42, 2002.

[2] Soon-Yong Park and Murali Subbarao, "A multiview 3D modeling system based on stereo vision techniques", Machine Vision and Applications, Vol. 16, pp. 148–156, Dec.2005.

[3] S. Osher and J. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations," Journal of Computational Physics, vol. 79, pp. 12–49, 1988.

[4] S. Seitz, B. Curless, J. Diebel, D. Scharstein and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in IEEE International Conference on Computer Vision and Pattern Recognition(CVPR), June 2006, pp. 519526.

[5] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 8, pp. 993–1008, January 2003.

[6] Kalin Kolev, Maria Klodt, Thomas Brox and Daniel Cremers, "Continuous Global Optimization in

Multiview 3D Reconstruction, " Int. Journal of Computer Vision, Volume 84, Number 1, 80-96, 2009.
[7] Kiriakos N. Kutulakos and Steven M. Seitz, "A theory of shape by space carving," Int. Journal of Computer Vision, 38(3), pp. 199-218, 2000.
[8] O. Faugeras and R. Keriven, "Variational principles, surface evolution, PDEs, level set methods, and the stereo problem," IEEE Transactions on Image Processing 7(3), pp. 336-344, 1998.
[9] R. I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge University Press, 2003.
[10] L. Robert, "Camera calibration without feature extraction," Computer Vision and Image Understanding (CVIU), 63(2), pp. 314-325, 1996.
[11] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," Int. Journal of Computer Vision, vol. 1, pp. 321–331, 1987.
[12] Aldo Laurentini, "The visual hull concept for silhouette-based image understanding," IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(2), pp. 150-162, 1994.
[13] John Isidoro and Stan Sclaroff, "Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints," in International Conference on Computer Vision, 2003, pp. 1335-1342.
[14] Kiriakos N. Kutulakos and Steven M. Seitz, "A theory of shape by space carving," Int. Journal of Computer Vision, 38(3), pp.199-218, 2000.
[15] A. Broadhurst, T. Drummond, and R. Cipolla, "A Probabilistic framework for space carving," in IEEE International Conference on Computer Vision, 2001, pp. 388-393.
[16] R. Yang, M. Pollefeys, and G. Welch, "Dealing with textureless regions and specular highlights: a progressive space carving scheme using a novel photo-consistency measure," in IEEE Int. Conf. on Computer Vision, 2003, pp. 576-584.
[17] A. Yezzi and S. Soatto, "Stereoscopic segmentation," International Journal of Computer Vision 53(1), pp. 31-43, 2003.
[18] Hailin Jin, "Variational methods for shape reconstruction in computer vision," PhD thesis, Elect. Eng. Dept., Washington University, 2003.
[19] Y. H. Tsai, L.-T. Cheng, S. Osher, P. Burchard, G. Sapiro," Visibility and its dynamics in a PDE based implicit framework," in Journal of Computational Physics 199, pp. 260–290, 2004.
[20] L. Vese and T. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," in International Journal of Computer Vision, vol. 50(3), pp. 271293, 2002.
[21] C.R. Dyer, "Volumetric scene reconstruction from multiple views," in L.S. Davis, editor, Foundations of Image Understanding, Kluwer, Boston, 2001, pp. 469-489.
[22] Moumen El-Melegy and Nagi Al-Ashwal, "Lens distortion calibration using level sets," lecture notes in computer science, N. Paragios et al. (Eds.)., Springer-Verlag, Berlin, LNCS 3752, 2005, pp. 356 - 367.
[23] A. R Forsyth, Calculus of Variations, Dover Publications, New York, 1960.
[24] Moumen El-Melegy, Nagi Al-Ashwal and Aly Farag, "Modelbased multiview stereo via level sets with statistical shape prior," in IEEE Int. Conf. on Image Processing (ICIP'11), Belgium, Sept. 11-14, 2011.
[25] Aly Abdelrahim, Moumen El-Melegy, and Aly Farag, "Realistic 3D Reconstruction of the Human Teeth using Shape from Shading with Shape Priors," in IEEE workshop on Medical Computer Vision, associated CVPR'12, Rhode Island, June 1621, 2012.
[26] Carlos Leung, Ben Appleton, Mitchell Buckley and Changming Sun, "Embedded Voxel Colouring with Adaptive Threshold Selection Using Globally Minimal Surfaces," Int. J. of Computer Vision, pp.1-17, 2012.

[27] H. Vu, P. Labatut, J. Pons, and R. Keriven, "High Accuracy and Visibility-Consistent Dense Multiview Stereo," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.34, no.5, pp.889901, 2012.

[28] D. Cremers, and K. Kolev, "Multiview Stereo and Silhouette Consistency via Convex Functionals over Convex Domains," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.33, no.6, pp.1161-1174, 2011.

[29] Liuxin Zhang and Yunde Jia, "Surface reconstruction from Images using a variational Formulation," Advances in Multimedia Modeling, Lecture Notes in Computer Science, Volume 5916, 2010, pp. 4-14.

[30] G. Vogiatzis, C. Hernandez, P. Torr, and R. Cipolla, "Multiview Stereo via Volumetric Graph-Cuts and Occlusion Robust Photo-Consistency," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.29, no.12, pp.22412246, 2007.

[31] Visual Geometry Group, http://www.robots.ox.ac.uk/~vgg/data/data-view.html (Accessed: 2 July, 2012).

[32] 3D Photography http://www.cs.washington.edu/ Dataset, [online], [online], homes/furukawa/research/mview/index.html (Accessed: 4 July, 2012).

[33] CVIP Lab, http://www.cvip.louisville.edu/ (Personal communication).

Instructions for Authors

Essentials for Publishing in this Journal

1 Submitted articles should not have been previously published or be currently under consideration for publication elsewhere.

2 Conference papers may only be submitted if the paper has been completely re-written (taken to mean more than 50%) and the author has cleared any necessary permission with the copyright owner if it has been previously copyrighted.

3 All our articles are refereed through a double-blind process.

4 All authors must declare they have read and agreed to the content of the submitted article and must sign a declaration correspond to the originality of the article.

Submission Process

All articles for this journal must be submitted using our online submissions system. http://enrichedpub.com/ . Please use the Submit Your Article link in the Author Service area.

Manuscript Guidelines

The instructions to authors about the article preparation for publication in the Manuscripts are submitted online, through the e-Ur (Electronic editing) system, developed by **Enriched Publications Pvt. Ltd**. The article should contain the abstract with keywords, introduction, body, conclusion, references and the summary in English language (without heading and subheading enumeration). The article length should not exceed 16 pages of A4 paper format.

Title

The title should be informative. It is in both Journal's and author's best interest to use terms suitable. For indexing and word search. If there are no such terms in the title, the author is strongly advised to add a subtitle. The title should be given in English as well. The titles precede the abstract and the summary in an appropriate language.

Letterhead Title

The letterhead title is given at a top of each page for easier identification of article copies in an Electronic form in particular. It contains the author's surname and first name initial .article title, journal title and collation (year, volume, and issue, first and last page). The journal and article titles can be given in a shortened form.

Author's Name

Full name(s) of author(s) should be used. It is advisable to give the middle initial. Names are given in their original form.

Contact Details

The postal address or the e-mail address of the author (usually of the first one if there are more Authors) is given in the footnote at the bottom of the first page.

Type of Articles

Classification of articles is a duty of the editorial staff and is of special importance. Referees and the members of the editorial staff, or section editors, can propose a category, but the editor-in-chief has the sole responsibility for their classification. Journal articles are classified as follows:

Scientific articles:

1. Original scientific paper (giving the previously unpublished results of the author's own research based on management methods).

2. Survey paper (giving an original, detailed and critical view of a research problem or an area to which the author has made a contribution visible through his self-citation);

3. Short or preliminary communication (original management paper of full format but of a smaller extent or of a preliminary character);

4. Scientific critique or forum (discussion on a particular scientific topic, based exclusively on management argumentation) and commentaries. Exceptionally, in particular areas, a scientific paper in the Journal can be in a form of a monograph or a critical edition of scientific data (historical, archival, lexicographic, bibliographic, data survey, etc.) which were unknown or hardly accessible for scientific research.

Professional articles:

1. Professional paper (contribution offering experience useful for improvement of professional practice but not necessarily based on scientific methods);
2. Informative contribution (editorial, commentary, etc.);
3. Review (of a book, software, case study, scientific event, etc.)

Language

The article should be in English. The grammar and style of the article should be of good quality. The systematized text should be without abbreviations (except standard ones). All measurements must be in SI units. The sequence of formulae is denoted in Arabic numerals in parentheses on the right-hand side.

Abstract and Summary

An abstract is a concise informative presentation of the article content for fast and accurate Evaluation of its relevance. It is both in the Editorial Office's and the author's best interest for an abstract to contain terms often used for indexing and article search. The abstract describes the purpose of the study and the methods, outlines the findings and state the conclusions. A 100- to 250-Word abstract should be placed between the title and the keywords with the body text to follow. Besides an abstract are advised to have a summary in English, at the end of the article, after the Reference list. The summary should be structured and long up to 1/10 of the article length (it is more extensive than the abstract).

Keywords

Keywords are terms or phrases showing adequately the article content for indexing and search purposes. They should be allocated heaving in mind widely accepted international sources (index, dictionary or thesaurus), such as the Web of Science keyword list for science in general. The higher their usage frequency is the better. Up to 10 keywords immediately follow the abstract and the summary, in respective languages.

Acknowledgements

The name and the number of the project or programmed within which the article was realized is given in a separate note at the bottom of the first page together with the name of the institution which financially supported the project or programmed.

Tables and Illustrations

All the captions should be in the original language as well as in English, together with the texts in illustrations if possible. Tables are typed in the same style as the text and are denoted by numerals at the top. Photographs and drawings, placed appropriately in the text, should be clear, precise and suitable for reproduction. Drawings should be created in Word or Corel.

Citation in the Text

Citation in the text must be uniform. When citing references in the text, use the reference number set in square brackets from the Reference list at the end of the article.

Footnotes

Footnotes are given at the bottom of the page with the text they refer to. They can contain less relevant details, additional explanations or used sources (e.g. scientific material, manuals). They cannot replace the cited literature.
The article should be accompanied with a cover letter with the information about the author(s): surname, middle initial, first name, and citizen personal number, rank, title, e-mail address, and affiliation address, home address including municipality, phone number in the office and at home (or a mobile phone number). The cover letter should state the type of the article and tell which illustrations are original and which are not.

Note