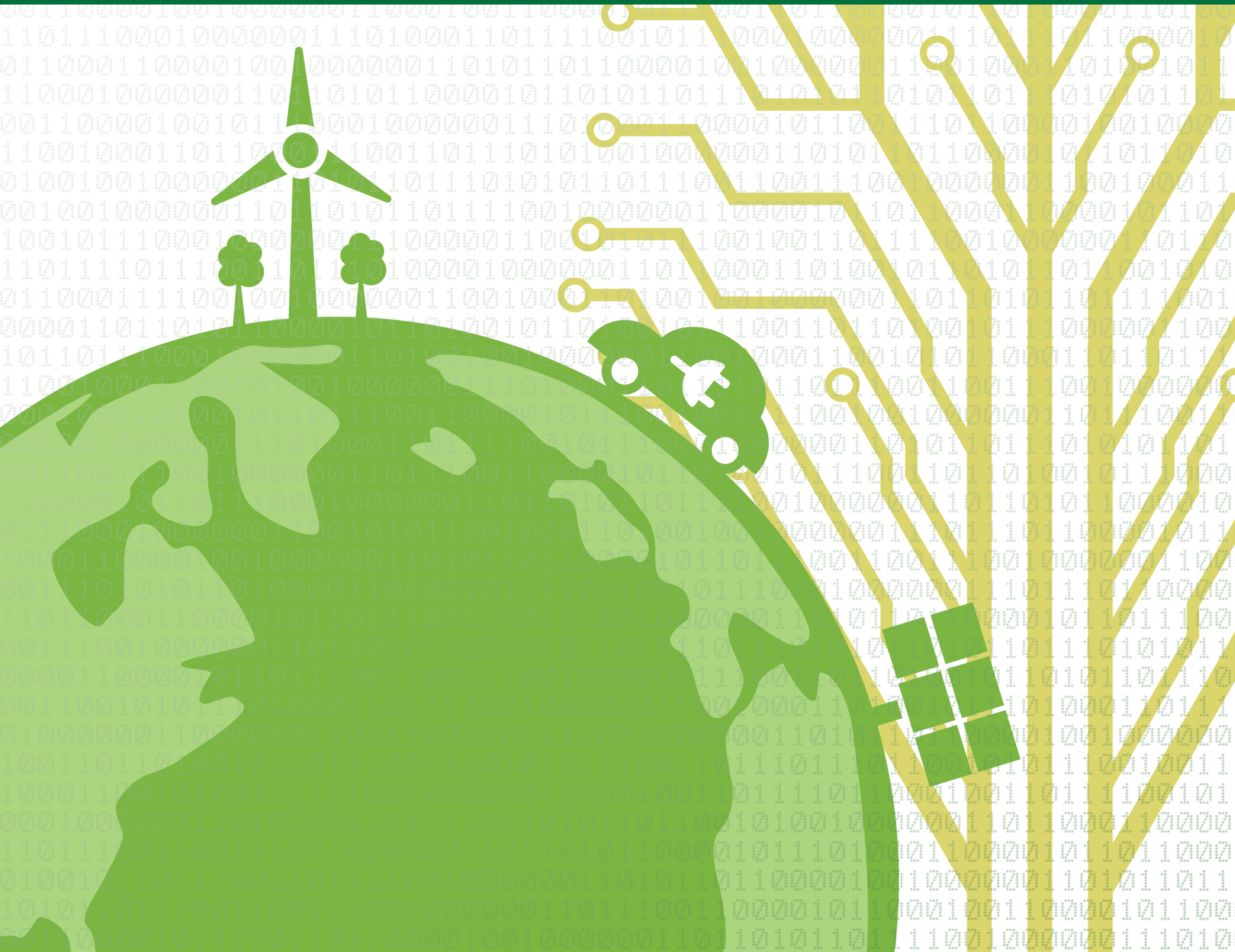
ISSN 2507-9174

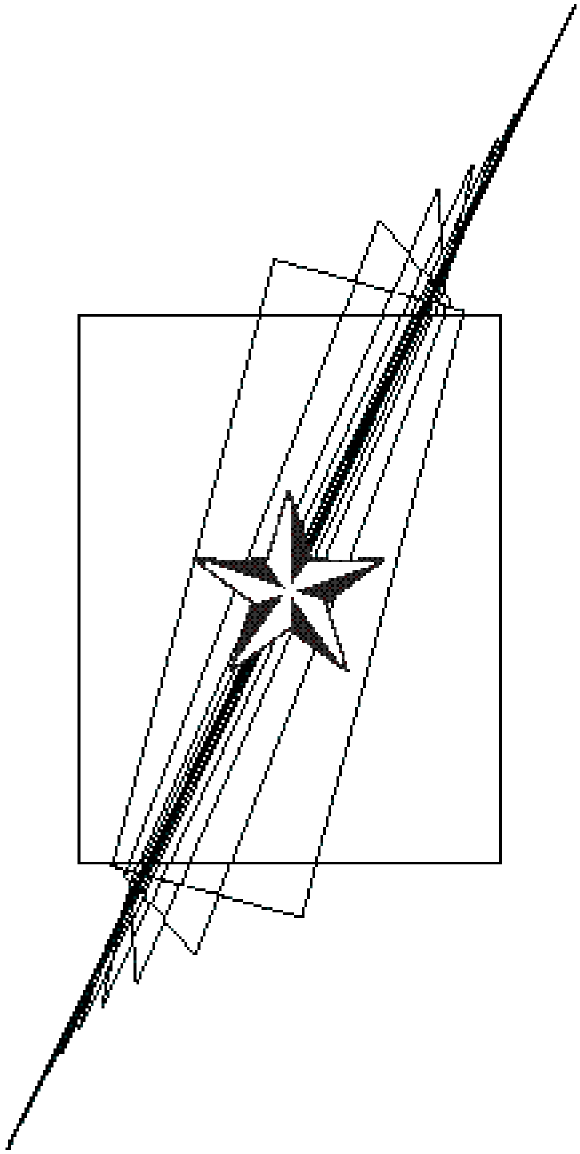
De La Salle University

Journal of Computational Innovations and Engineering Applications

Vol.1 No. 2

January 2017





JOURNAL OF COMPUTATIONAL INNOVATIONS AND ENGINEERING APPLICATIONS

Volume 1 Number 2
JANUARY 2017

The **Journal of Computational Innovations and Engineering Applications (JCIEA)** is a peer-reviewed, open access journal of De La Salle University, Manila. The JCIEA aims to promote the development of new and creative ideas on the use of technology in solving different problems in different fields of our daily lives. The JCIEA solicits high quality papers containing original contributions in all areas of theory and applications of Engineering and Computing including but not limited to: Computational Applications, Computational Intelligence, Electronics and Information and Communications Technology (ICT), Manufacturing Engineering, Energy and Environment, Robotics, Control and Automation, and all their related fields. The JCIEA editorial board is comprised of experts from around the world who are proactively pushing for the development of research in these fields.

Annual Subscription Rates: Foreign libraries and institutions: US$60 (airmail). Individuals: US$50 (airmail). Philippine domestic subscription rates for libraries and institutions: Php1,800, individuals: Php1,300. Please contact Ms. Joanne Castañares for subscription details: telefax: (632) 523-4281, e-mail: dlsupublishinghouse@dlsu.edu.ph

ISSN 2507-9174

Published by De La Salle University Publishing House
2401 Taft Avenue, Manila 0922 Philippines
Telephone: (63 2) 523-4281 / 524-2611 loc 271
Fax: (63 2) 523-4281
Email: dlsupublishinghouse@dlsu.edu.ph
Website: http://www.dlsu.edu.ph/offices/publishing-house/journals.asp

*The De La Salle University Publishing House is the publications office
of De La Salle University, Manila, Philippines.*

JOURNAL OF COMPUTATIONAL INNOVATIONS AND ENGINEERING APPLICATIONS

Table of Contents

From the Editor

Research Articles

From the Editor

The Journal of Computational Innovations and Engineering Applications (JCIEA) is a peer-reviewed and abstracted journal published twice a year by De La Salle University, Manila, Philippines. JCIEA aims to promote and facilitate the dissemination of quality research outputs that can push for the growth of the nation's research productivity.

In its second issue, 7 articles are selected, which will provide valuable references for researchers and practitioners.

The first article is "Predicting the Quality of Demosaiced Images Using the Sparsity of Chroma Gradients." This paper explores a no-reference simple metric for inferring the quality of the estimated image by measuring the sparsity of chroma gradients along four directions (SCG4). The said measure is shown to be significantly correlated with respect to the PSNR in simulations using the Kodak image database.

The second article, "Development and Implementation of an Integrated Services Digital Broadcasting-Terrestrial (ISDB-T) Decoder Using GNU Radio and USRP1," discussed the implementation of an ISDB-T decoder in a Software-Defined Radio (SDR) platform.

The third article, "Efficient Load Balancing Technique for Parallel Ray Tracing Using A Reservoir," proposed an alternative load balancing technique to further increase computing efficiency and improve ray tracing's applicability for mainstream purposes. The load balancing technique that uses a reservoir was implemented to tackle the issue of using heterogeneous computers in the network and to minimize the communication overhead introduced by parallel applications.

The fourth article is "Analog Realization of a Low-Voltage Sixteen Selectable Fractional-Order Differentiator in a 0.35um CMOS Technology." This paper focused on the design and implementation of sixteen selectable fractional-order differentiators in a 0.35um CMOS technology operated at 1.5-V supply. In comparison with the previous work that uses generic microcontroller for switching an FOD from one order to the next, this design of a 16 selectable FOD was realized in an analog microelectronic scale, thus, the physical implementation is relatively smaller.

The fifth article, "Comparison of Logit and Neural Network Models in Inter-Island Discrete Choice Analysis," wherein the researchers addressed the non-linear behavior and inter-dependence of variables using neural networks in modeling inter-island travel choice. The neural network model is statistically acceptable in describing travel choice behavior, while the logit model is more inclined to model the decision-making process.

The sixth article, “Selection of Artificial Neural Network Training Algorithms in the Detection and Classification of Wavelet de-noised Musical Tone Stimulated EEG Signals,” musical tones were used to stimulate the brain and EEG signals are used to detect and classify these stimulations. Artificial Neural Network-based classifier was employed for detection and classification. Wavelet based de-noising was used to smoothen the musical tone stimulated EEG signals and among the 110 known mother wavelets, the reverse biorthogonal ‘rbio3.1’ and ‘rbio3.3’ using the ‘rigrsure’ thresholding method satisfied the selection criteria for better de-noising effects.

The seventh article, “Fuzzy Implementation for MCU on Power Savings and Efficient Irrigation System (MPSEIS) for Smart Farming,” the researchers present a self-learning controller for motor speed to be utilized on Three Phase Motor using Variable Frequency Driver (VFD) for irrigation system of Smart Farming using fuzzy logic algorithm developed inside a Micro-Control Unit (MCU) environment. To test the controller’s performance, different frequencies using variable frequency driver (VFD) in real time undergoing different water level and power load variations.

The JCIEA editorial board expresses their warmest thanks and deepest gratitude to the distinguished authors for their outstanding contribution to JCIEA 2nd volume. They likewise express profound appreciation to the reviewers for their assistance and cooperation. Original research outputs are most welcome to JCIEA. There is no publication fee in this journal, and the research papers are assured of fair and fast peer review process. For further information, please visit www.dlsu.edu.ph/offices/publishinghouse/journals.asp.

Prof. Elmer P. Dadios, PhD
Editor-in-Chief, JCIEA

Predicting the Quality of Demosaiced Images Using the Sparsity of Chroma Gradients

Carlo Noel Ochotorena, Cecille Adrianne Ochotorena, and Yukihiko Yamashita

***Abstract*—The design of most modern cameras utilizes a color filter array that downsamples and interleaves the red, green, and blue pixels of an image into a single mosaiced image. Such a design makes it necessary to interpolate the missing pixels for each color channel using a process known as demosaicing. While it is possible to fill in these pixels, the resulting images are inexact estimates of the true image, with different algorithms offering various levels of success. However, this degree of success cannot be directly quantified in the absence of the true image, making it difficult to design adaptive algorithms for demosaicing. This paper explores a no-reference simple metric for inferring the quality of the estimated image by measuring the sparsity of chroma gradients along four directions (SCG4). The said measure is shown to be significantly correlated with respect to the PSNR in simulations using the Kodak image database.**

***Index Terms*—demosaicing, color filter array, gradient, sparsity.**

I. Introduction

WITH the growing presence of imaging systems in the modern world, it is not surprising that the underlying technology behind such imaging devices have received a proportional amount of attention. In particular, digital imaging sensors have continuously been developed through the years. One of the most apparent aspects of this growth is seen in the resolution of the imaging sensors. As consumer video is pushing for 4K video resolutions and higher, sensor technology has to cope with the market demand. Alongside the increasing resolution [1], [2], [3], there is also a growing commercial interest in high speed video captures [4], [5], [6].

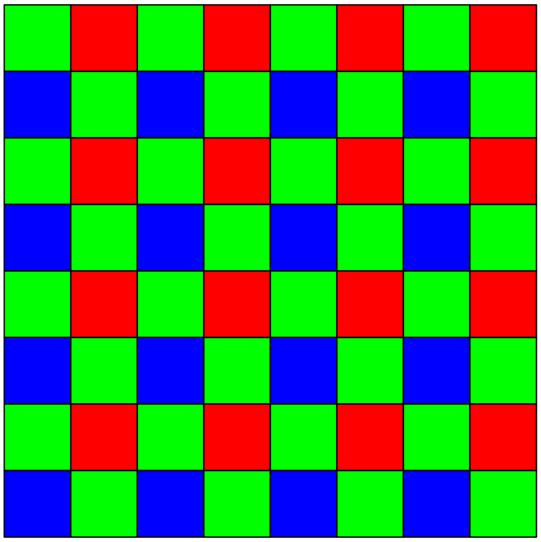


Fig. 1. Bayer color filter array (CFA). Each color site is comprised of a color filter (typically a chemical dye) placed in from of a photodetector.

Driven by the progressively rising requirements, researchers have come up with ways to improve sensor technologies. For instance, the construction of smaller sensors have enabled consumer devices such as mobile phones to capture high-resolution images and video [1], [2], [3]. Sensors with fast readout capabilities can now be utilized to capture high framerate videos (e.g. 1500 fps [6], 10000 fps [5], etc). Improving sensitivities have allowed images and videos to be captured under poor lighting conditions [7], [8]. In a similar manner, developments in sensitivity have also allowed for the lower levels of noise in captured images.

Despite the changes in sensor design throughout the years, one aspect of imaging sensors has remained vastly unchanged since the 1970s—the manner by which color is captured. At the very core of most imaging sensors is an array of photosensitive devices designed to capture incident light and translate the intensity into electronic signals [9]. However, these devices can only describe the intensity of

C. N. Ochotorena is a Ph.D. candidate of International Development Engineering at the Tokyo Institute of Technology, Japan (e-mail: carlo.ochotorena@dlsu.edu.ph)

C. A. Ochotorena is a Ph.D. candidate of Electronics and Communications Engineering at the De La Salle University, Philippines (e-mail: cec.ochotorena@dlsu.edu.ph)

Y. Yamashita is with the International Development Engineering Department, Tokyo Institute of Technology, Japan

light captured over a broad spectrum and not for specific wavelengths, thus making them incapable of quantifying the intensity of different colors.

To address this limitation, wavelength-specific filters are placed in front of the individual sensor cells to allow them to capture color-specific intensities. Collectively, these filters form what is known as a color filter array (CFA). A prominent example of this is the Bayer CFA [9] (shown in Figure 1), which is still used in many imaging sensors today. The downside to such an approach is that each cell in the array can only measure one particular color, effectively downsampling the individual color channels. The resulting captured image appears to be monochromatic (see Figure 2) as only a single flat image is obtained from the sensor. However, this flat image actually represents the interleaving of the intensities of three color channels.

As a result of the interleaving of color channels, the resulting image is often not in a usable form and has to be deinterleaved to form the individual color channels. This leaves missing pixels for each color channel which has to be interpolated to reconstruct the colored image. The interpolation process for images obtained through a CFA is specifically known as demosaicing and is generally a non-trivial reconstruction task as information is readily lost during the downsampling process. For this reason, demosaicing has been a subject of interest to many researchers since the conception of the CFA [11].

As with almost any reconstruction task, the true image, and subsequently the quality of the reconstruction, is unknown to the process. If such quality information were made available, an "oracle" process would be able to make optimal decisions during the reconstruction. Using this premise, this work proposes a no-reference metric that predicts the quality of demosaicing in the absence of the true image. By utilizing such a measure, more effective demosaicing algorithms can be designed. To understand how such a metric can be developed, we first introduce Bayer CFA along with some technical aspects of this array in Section II. An overview of some demosaicing paradigms is provided in Section III. Following this, the proposed metric is discussed in Section IV along with some experiments in Section V.

II. The Bayer CFA

While there have been many color filter array (CFA) designs proposed, the Bayer CFA is still one of the most widely used patterns today [9], [11] This CFA (as illustrated in Figure 1) creates a repeated array of red, green, and blue filters in such a way that every 2×2 pixel area of the entire array contains exactly two green pixels and one each of the red and blue pixels. This construction is consistent with the observation that the human eyes are generally more sensitive to green wavelengths thus making the spatial resolution of the green channel more significant towards our perception of visual quality. To describe the Bayer CFA, we begin by defining the intensities of the red, green, and blue color channels of an image as $f_{\mathrm{R}}(x, y)$, $f_{\mathrm{G}}(x, y)$, and $f_{\mathrm{B}}(x, y)$, respectively for a given pixel coordinate (x, y). Using this notation, any CFA can be described using three masks corresponding to the color channels—$m_{\mathrm{R}}(x, y)$, $m_{\mathrm{G}}(x, y)$, and $m_{\mathrm{B}}(x, y)$. Using these masks, the mosaiced image obtained by the sensor is described as:

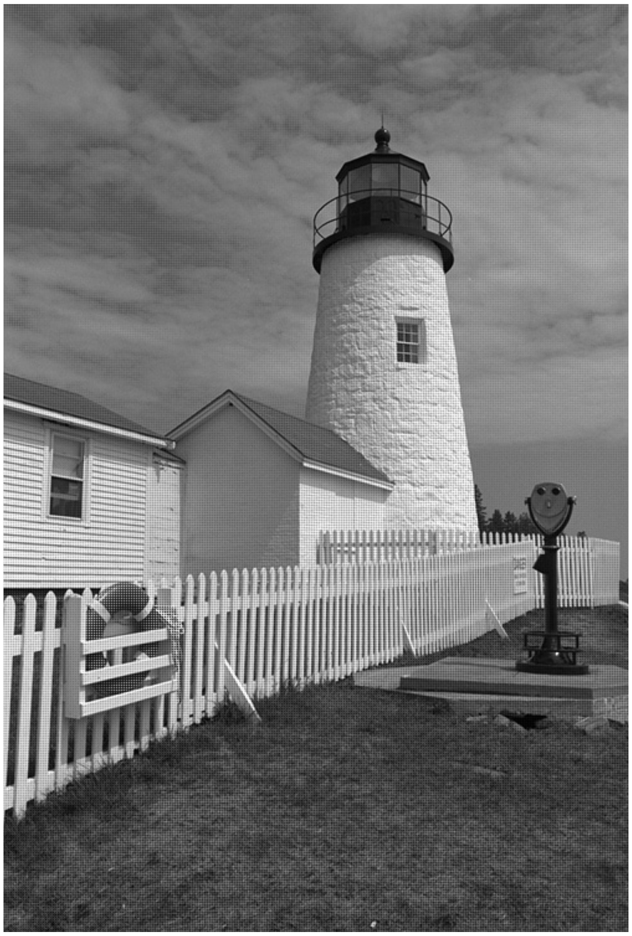


Fig. 2. A mosaiced image of the *Lighthouse* image from the *Kodak* color image database [10] passed through a simulated Bayer CFA.

$$\begin{aligned} f_{\mathrm{CFA}}(x, y) = & f_{\mathrm{R}}(x, y) m_{\mathrm{R}}(x, y) + \\ & f_{\mathrm{G}}(x, y) m_{\mathrm{G}}(x, y) + \\ & f_{\mathrm{B}}(x, y) m_{\mathrm{B}}(x, y) \end{aligned} \tag{1}$$

With slight abuse of notation, we omit the coordinates for the individual channels in the succeeding discussion for brevity thus using the expressions f_{CFA}, f_{R}, f_{G}, and f_{B} instead.

In the specific case of the Bayer CFA, these masks are position dependent and repeat for every 2 × 2 region of the grid. While there are variations in the phase convention for the Bayer array, in this work, we followed the convention defined in [12] and define the masks as:

$$m_{\mathrm{R}}(x,y) = \begin{cases} 1, & x \text{ is odd and } y \text{ is even} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$m_{\mathrm{G}}(x,y) = \begin{cases} 1, & x+y \text{ is even} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$m_{\mathrm{B}}(x,y) = \begin{cases} 1, & x \text{ is even and} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

A more mathematical way of expressing (2) – (4) is described by [12] in the form:

$$m_{\mathrm{R}}(x,y) = \frac{1}{4}\left[1-(-1)^{x}\right]\left[1+(-1)^{y}\right] \tag{5}$$

$$m_{\mathrm{G}}(x,y) = \frac{1}{2}\left[1+(-1)^{x+y}\right] \tag{6}$$

$$m_{\mathrm{B}}(x,y) = \frac{1}{4}\left[1+(-1)^{x}\right]\left[1-(-1)^{y}\right] \tag{7}$$

Combining (1) with (5) – (7) leads to the expression:

$$\begin{aligned} f(x,y) =& \frac{1}{4} f_{\mathrm{R}}\left[1-(-1)^{x}\right]\left[1+(-1)^{y}\right] + \\ & \frac{1}{2} f_{\mathrm{G}}\left[1+(-1)^{x+y}\right] + \\ & \frac{1}{4} f_{\mathrm{B}}\left[1+(-1)^{x}\right]\left[1-(-1)^{y}\right] \end{aligned} \tag{8}$$

which can subsequently be expanded and regrouped to form:

$$\begin{aligned} f(x,y) = & \left[\frac{1}{4} f_{\mathrm{R}} + \frac{1}{2} f_{\mathrm{G}} + \frac{1}{4} f_{\mathrm{B}}\right] + \\ & \left[-\frac{1}{4} f_{\mathrm{R}} + \frac{1}{2} f_{\mathrm{G}} - \frac{1}{4} f_{\mathrm{B}}\right](-1)^{x+y} + \\ & \left[-\frac{1}{4} f_{\mathrm{R}} + \frac{1}{4} f_{\mathrm{B}}\right]\left[(-1)^{x} - (-1)^{y}\right] \end{aligned} \tag{9}$$

The first term in the above expression is roughly analogous to a luma channel, f_L, while the remaining two terms can be interpreted as chroma channels, f_{C1} and f_{C2}, multiplied by distinct modulation functions. These individual components can be described as:

$$f_{\mathrm{L}} = \frac{1}{4} f_{\mathrm{R}} + \frac{1}{2} f_{\mathrm{G}} + \frac{1}{4} f_{\mathrm{B}} \tag{10}$$

$$f_{\mathrm{C1}} = -\frac{1}{4} f_{\mathrm{R}} + \frac{1}{2} f_{\mathrm{G}} - \frac{1}{4} f_{\mathrm{B}} \tag{11}$$

$$f_{\mathrm{C2}} = -\frac{1}{4} f_{\mathrm{R}} + \frac{1}{4} f_{\mathrm{B}} \tag{12}$$

This formulation implies that the mosaiced image is essentially a grayscale image (i.e., luma component) corrupted by “noise” from the chroma components. If the chroma components are fully known, the original image can easily be reconstructed from the mosaiced image. For this reason, the chroma components play a pivotal role in our proposed metric.

III. DEMOSAICING OVERVIEW

Equipped with a fair understanding on the design of the Bayer CFA, the next task is to design a system for reconstructing the original image from the mosaiced samples. The most trivial approach to this problem is to simply interpolate the missing pixels of each channel given the known pixels. Since the red and blue channels are effectively downsampled to half of their resolution, image interpolation techniques can be applied to restore the resolution. The green channel, on the other hand, is sampled in a quincunx pattern and has more samples available, thus making interpolation more effective. A clear downside to naïve interpolation is that the high frequency components of the red and blue channels that were discarded during the downsampling process cannot be restored upon doubling [11].

Beyond simply the loss of information, handling each color channel independently leads to more problematic artifacts in the image. Take, for instance, the well-known *Lighthouse* image from the *Kodak* database (see Figure 3). After mosaicing this image and subsequently applying a bicubic interpolant to each color channel, some visible color bands become apparent in the reconstructed image. These bands, often referred to as color moiŕe, result from incorrect and out-of-phase interpolation decisions. An early attempt to

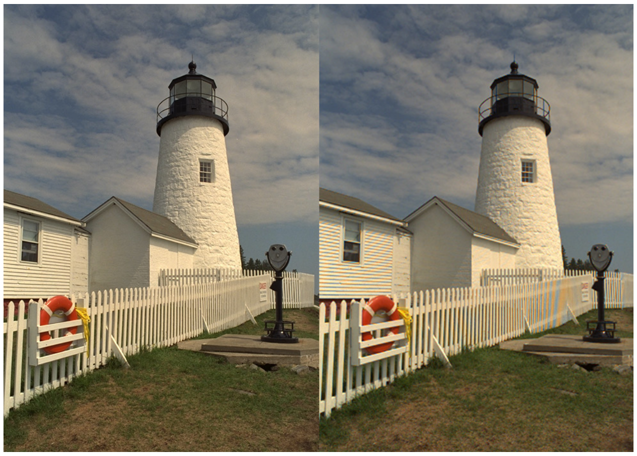


Fig. 3. Reconstruction of a mosaiced image using bicubic interpolation. Naive interpolation can lead to color artifacts in the reconstructed image (right) that are not present in the original image (left).

address such artifacts is to adaptively alter the interpolation direction based on horizontal and vertical gradients of the color channels [11]. This allows the interpolation to adapt to edges in the underlying image to achieve better results. Similar approaches can adapt the weights of neighboring pixels to avoid edge discontinuities [13].

Another demosaicing paradigm is to exploit the relationship between color channels. Since color channels in images are naturally correlated with one another, many demosaicing techniques utilize this relationship to obtain better interpolation results [13], [14], [15], [16], [11], [17], [18]. For instance, many approaches operate under the assumption that hues in an image are relatively slow-changing allowing the differences or ratios between color channels to be utilized for prediction [13], [14], [15], [11]. Other approaches interpolate the green channel independently and use it to guide the red and blue interpolation [16], [17], [18]. A state-of-the-art, Minimized-Laplacian Residual Interpolation (MLRI), tackles the problem using a similar perspective by using guided upsampling process with the interpolated green channel to generate an initial estimate for the red and blue channels [17]. This estimate is then updated using an approximate Laplacian minimization criteria to arrive at a refined solution.

Aside from relying on color differences, some techniques also use adaptive fusion of different estimates of the image. A popular algorithm, known as Adaptive Homogeneity-Directed (AHD) demosaicing [19], performs both horizontal and vertical interpolations of the color channels and merges the two estimates based on a homogeneity metric designed to indicate the presence of color artifacts. Using a very different approach, the frequency-domain approach [12] models the chroma modulation process (discussed in Section II) in the Fourier domain. In this domain, it becomes apparent that color artifacts appear from crosstalk between the luma channel f_{L} and the second chroma channel f_{C2}. The authors of the said work proposed using a non-adaptive least-squares (LS) filter to extract the first chroma component f_{C1} and two non-adaptive LS filters to estimate f_{C2} based on the horizontally and vertically modulated contributions to the image. This results in two images which, similar to AHD, are adaptively merged to form the final image.

Despite the wide variations in approach of these different works, a common feature for many of them is that the optimization criteria is often carried out in the ℓ_2 domain. While this offers a simplistic solution to the problem, many properties of natural images are inherently sparse and not sufficiently served by the ℓ_2 domain. In the succeeding section, we highlight some of the said properties and propose a potential improvement to this criteria by exploiting such sparsity.

IV. Gradient Sparsity

A. Properties of Natural Images

In order to develop our metric, it is first useful to characterize certain properties of natural images. However, as the term “chroma” is used in an ambiguous manner in literature and may refer to different mixtures of the color channels, we explicitly present a series of experiments carried out using the 24 images of the *Kodak* color image database. This allows us to disambiguate the chroma definition and focus on the properties of the specific variant of chroma defined in equations (11) and (12). For these experiments, each of the 24 images are mapped into their respective luma and chroma components, thus generating a set of ground truths which are then characterized.

1) Sparsity of Gradients: One of the central features of our proposed metric is the use of gradients of components. To be specific, we define the gradient in four directions of an arbitrary component f as follows:

$$\nabla_{-x} f \equiv f(x,y) - f(x-1,y) \tag{13}$$

$$\nabla_{+x} f \equiv f(x,y) - f(x+1,y) \tag{14}$$

$$\nabla_{-y} f \equiv f(x,y) - f(x,y-1) \tag{15}$$

$$\nabla_{+y} f \equiv f(x,y) - f(x,y+1) \tag{16}$$

Using the above definitions, we calculate the gradients in four directions for all pixels in the *Kodak* database. To make these measures more meaningful, we obtain the histograms of each gradient direction and each component with a bin size of 1. The resulting plots, as seen in Figure 4, make it apparent that the majority of chroma gradients take on a zero value. The remaining non-zero gradients are still highly likely to be close to zero. On the other hand, the luma component does not exhibit the same behavior. Here, many of the gradients take nonzero values. This test demonstrates how the chroma components of natural images are *approximately* sparse (i.e., contain few large values).

2) Sparsity of Gradients in Four Directions: While the assertion that the gradient of chroma components are sparse, is useful in itself, we delve further into the properties of this sparsity. The previous experiment focused primarily on the individual gradient components without considering how these interact. Looking back at our definition of the four gradients (equations (13)–(16)), we find that these gradients are centered around a given pixel. We make use of this by counting the number of significant gradients around that pixel.

In this second experiment, we consider that every pixel can have between 0 to 4 non-zero gradients associated with it. Previously, we have established that the chroma

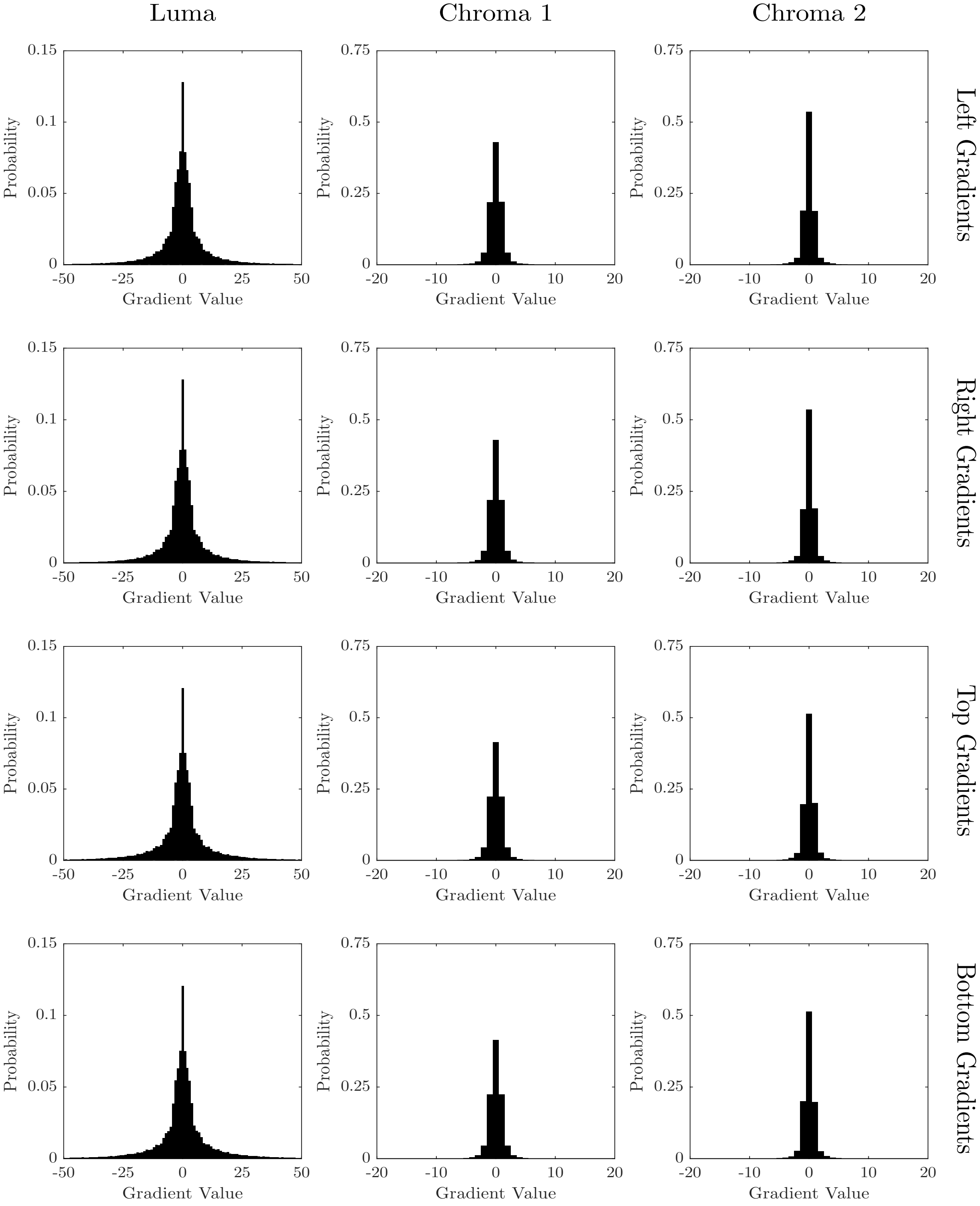


Fig. 4. Histogram of image gradients taken along four directions of various image components. Generally, the probabilities of a zero gradient are significantly higher for that of the chroma components compared to that of the luma component, regardless of the gradient direction.

components are *approximately* sparse due to factors such as noise. As such, a simple criteria such as dividing gradients between zeros and nonzeros is not necessarily reliable. Instead of applying a fixed threshold of 1, we model the probabilistic occurrence around a certain threshold. For each pixel center, the number of gradient magnitudes above or equal to the threshold is recorded. The probabilities of each number (from 0 to 4) are computed from all pixels accumulated from the *Kodak* database. The stacked bar graph in Figure 5 illustrates the trend at various threshold levels.

This test demonstrates how likely the gradients in four directions around a pixel are significant. In either chroma component, it is again apparent that most pixels have insignificant (i.e., below a given threshold) gradients. More importantly, this makes it clear that for a given pixel, there is a high probability that only one of the four gradients are of significant magnitude.

3) Correlating Gradients between Chroma Components: Having established a sparsity trend for each pixel, it is possible to apply the said observation towards developing our metric. However, there is one more property of natural images that is of some interest. An intuitive notion for images is that an edge occurring inside a given region often accompanies a change in color. If this were to hold true, such an edge would present itself in all three components of an image. To verify this, we consider the gradient magnitude at each direction. For each discretized magnitude in the first chroma component, we find the resulting distribution on the second chroma component. This can be visualized using the box plot in Figure 6.

If there exists no correlation between the gradients in two chroma components, we would expect the entire plot to remain close to zero (based on the observations from the previous experiment). The box plots obtained from the sample data, however, clearly demonstrate rising median and quartile levels as the chroma 1 gradient magnitude increases. Such a trend is consistent regardless of the gradient direction being studied. This test establishes that there is, indeed, some form of structural correlation between the two chroma components.

4) Summary: Given the results of the three experiments conducted on natural images, we have arrived at several key observations as summarized below:

1) Gradients of the chroma components are *approximately* sparse;
2) For a given pixel, there is a high probability that, at most, one of the four gradient directions are significant;
3) When a gradient in one of the four directions is significant in one chroma component, there is a non-negligible chance that it is also significant in the other.

B. The Proposed Metric

1) Reconstruction Quality of Luma-Chroma Components: For a predictive metric to be useful, it should not be dependent on any knowledge of the target image. This premise runs contradictory to the properties discussed in the previous section as these are dependent on the complete luma and chroma components. Furthermore, the use of the Bayer array prevents the calculation of any single gradient quantity from the mosaiced image. In order to tackle this limitation, in this work, we reconstruct one of the components and use it as *a priori* information for measuring the quality of the reconstruction.

As there are several components for a given image, a natural question that arises is: which one is suitable for use *a priori*? To answer this we characterize the reconstruction quality of each of the three components over different demosaicing methods. In particular, we use bilinear interpolation, bicubic interpolation, adaptive homogeneity directed (AHD) demosaicing [19], alternating projections (AP) [16], [20], Contour Stencils (CS) [21], [22], Directional Linear Minimum Mean Square-Error (DLMMSE) [23], [24], Least-squares Luma-Chroma Demosaicing (LSLCD) [12], Malvar-He-Cutler (MHC) demosaicing [25], [26], Minimized-Laplacian Residual Interpolation (MLRI) [17], and Successive Approximation (SA) [27]. Running each of these demosaicing algorithms across all the *Kodak* images, we arrive at various PSNR measures for the luma and chroma components. The medians of these measures can be seen in Figure 7.

An apparent observation from this test is that the chroma channels are better preserved in the reconstruction. Furthermore, the first chroma channel is typically reconstructed with higher fidelity. This is consistent with the assertion in [12] that this particular component suffers from minimal crosstalk by being modulated at a high frequency for both the horizontal and vertical axis. This choice of component is also convenient as the first chroma component can easily be approximated using the least-squares filters developed in [12] which can be applied through simple convolution.

2) Sparsity of Chroma Gradients: Having chosen a component to use *a priori*, we proceed to developing our proposed metric, the Sparsity of Chroma Gradients in Four Directions (SCG4). In the previous section, we assert that there exists a correlation between the occurrence of gradients of the two chroma components. This would intuitively lead to the notion that we can model the probability of significance of one component with respect to the significance of the other. However, doing so will enforce a strict measure along each gradient direction. We relax this criteria by, instead,

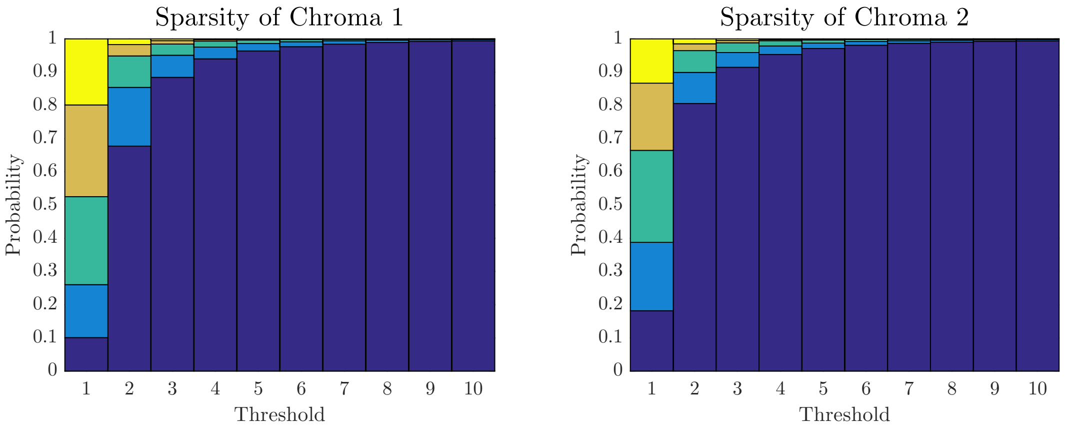


Fig. 5. Probability distribution of gradient magnitudes for each chroma channel above or equal to a given threshold. Since there are four gradient directions, each pixel in a given chroma component can have 0 (shown in blue) to 4 (shown in yellow) significant gradients occuring at different probabilities.

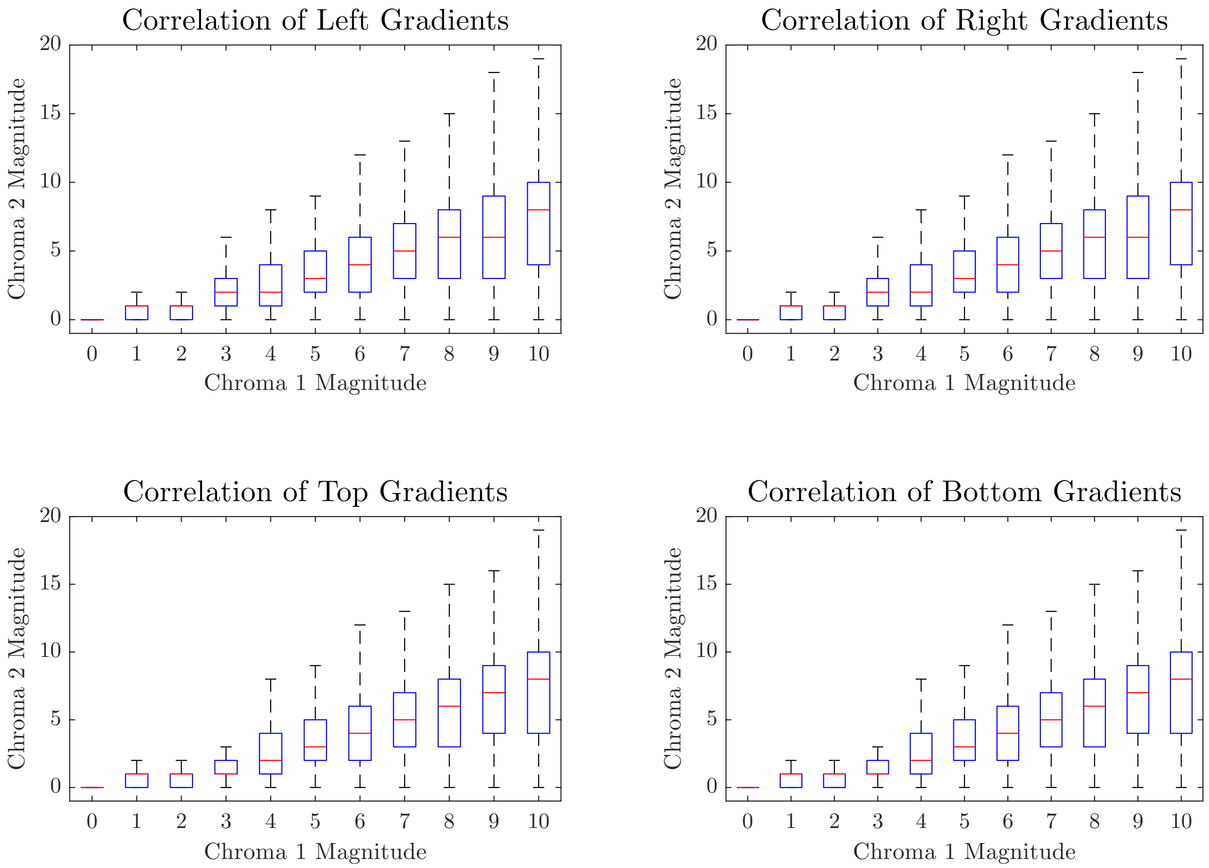


Fig. 6. Probability distribution of the gradients of the second chroma component at different magnitudes of the first chroma component. This box plot demonstrates that, regardless of the gradient direction, there exists a notable correlation between the gradients of the two chroma components for the medians (red line) and quartiles (blue box). It should also be noted that when the first chroma component has a magnitude of 1, the corresponding second chroma component has a very high probability of being zero resulting in a box plot with no visible quartiles and limits.

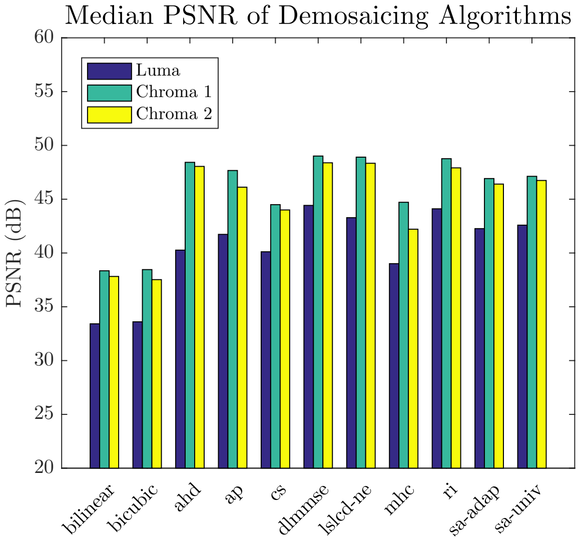


Fig. 7. Median PSNR of the luma and chroma components for the Kodak color image database.

measuring the number of significant gradients in each component defined as:

$$\mathcal{N}_t(x,y) = \mathcal{T}_t\left(|\nabla_{-x} f|\right) + \mathcal{T}_t\left(|\nabla_{+x} f|\right) + \mathcal{T}_t\left(|\nabla_{-y} f|\right) + \mathcal{T}_t\left(|\nabla_{+y} f|\right) \tag{17}$$

where the thresholding function $\mathcal{T}_t(x)$ is defined as:

$$\mathcal{T}_t(x) = \begin{cases} 0, & x < t \\ 1, & x \geq t \end{cases} \tag{18}$$

for an arbitrary threshold t. Using this measure, we can probabilistically model the relationship between two components. Since we use the first chroma as *a priori* information, we reconstruct it directly from the CFA by filtering and count the number of significant gradients for each pixel. We can then obtain the conditional probability of a certain number of significant gradients in the second chroma component exceeding that of the first:

$$p_{\mathrm{C2}}(n) \equiv p\left(\mathcal{N}_{\mathrm{C2}}(x,y) \geq n \mid \mathcal{N}_{\mathrm{C1}}(x,y) = n\right) \tag{19}$$

Such a probability is of interest in this work due to the correlation between the chroma components. If the second chroma component has more significant gradients than the first, we hypothesize that these gradients are more likely to be caused by color artifacts rather than actual image structures. We enforce this idea using empirical testing in the next section.

From a training perspective, these probabilities can readily be obtained from observations made between the filter-reconstructed C1 component, and the ground truth C2 component. Once the probabilities are obtained from the training set, they can simply be stored as a lookup-table for use in our metric. The metric itself is based on observations using a filter reconstructed C1 and the C2 component derived from the demosaiced image. This can be described using the following steps:

1) Given the CFA, use the filter from [12] to estimate the C1 component;
2) Given the demosaiced image, compute the C2 component using equation (12);
3) Calculate the four gradients at each pixel in the C1 and C2 components;
4) Determine the number of significant gradients around each pixel in each component;
5) Find the probability for each observed C2 count given the observed C1 count using equation (19);
6) The SCG4 metric is defined as the mean probability across all pixels in the image.

By using mean probabilities, we ensure that the final metric stays within the range of 0 to 1. A higher mean probability indicates a stronger correlation between the two chroma components and thus a better estimated quality. Conversely, a metric closer to zero indicates a C2 component that is inconsistent with the observations from the C1 component. In the succeeding section, we describe several tests that were used to verify the proposed metric.

V. Experiments

A. *Subjective Quality*

The first experiment conducted to verify the usefulness of the proposed metric is to evaluate it on a particularly

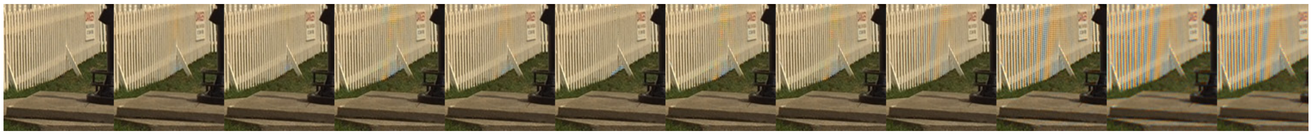


Fig. 8. A cropped region of the Lighthouse image sorted in descending order according to the SCG4 metric. Listed from left to right are the demosaiced images obtained using LSLCD [12], DLMMSE [23], [24], AP [16], [20], RI [17], AHD [19], SA-Universal and SA-Adaptive [27], CS [21], [22], MHC [25], [26], Bicubic interpolation, and Bilinear interpolation, respectively.

problematic region within the *Lighthouse* image from the *Kodak* database. To accomplish this, the probabilities were first obtained using the images statistics of all the images that are part of the *Kodak* database except for the *Lighthouse* image. This avoids self-training and is intended to strengthen the validity of the experiment.

Given the image statistics learned during the training process, we calculate the metric for the different demosaicing methods discussed in the previous section. In particular, the metric is only applied to a localized region within the Lighthouse image where a portion of the fence contains high-frequency information that is known to result in severe color moiŕe. These sub-images, sorted by the SCG4 metric, can be seen in Figure 8. While there are some deviations in the perceptual order, this figure shows that there is a subjective correlation between the visual quality and the proposed SCG4 metric.

B. Objective Quality with Cross-validation

Expanding on the methodology from the previous test, we obtained objective results from the *Kodak* color image database. Following a cross-validation procedure, each of the 24 images in the database are evaluated using statistics obtained from the other 23 images (excluding the image under test). Each of the demosaiced images are then objectively evaluated with respect to their original images using the peak signal-to-noise ratio (PSNR), a metric

TABLE I
Correlation Coefficients of the 24 Kodak Images at Various Thresholds

Kodak Image Number	**Significance Threshold (t)**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	**0.468**	0.940	0.961	0.971	0.970	0.959	0.945	0.930	0.920	0.920
2	**-0.817**	0.866	0.888	0.890	0.885	0.879	0.848	0.768	0.735	0.735
3	**-0.902**	0.915	0.925	0.928	0.923	0.920	0.920	0.920	0.918	0.918
4	**-0.829**	0.918	0.946	0.962	0.969	0.971	0.974	0.977	0.977	0.977
5	**-0.705**	0.931	0.953	0.969	0.977	0.981	0.981	0.979	0.975	0.975
6	**-0.755**	0.948	0.970	0.975	0.972	0.964	0.954	0.945	0.935	0.935
7	**-0.920**	0.918	0.946	0.953	0.958	0.960	0.955	0.950	0.945	0.945
8	**0.002**	0.943	0.957	0.966	0.972	0.972	0.967	0.960	0.952	0.952
9	**-0.900**	0.952	0.965	0.967	0.964	0.960	0.957	0.950	0.940	0.940
10	**-0.883**	0.961	0.974	0.976	0.976	0.975	0.975	0.975	0.974	0.974
11	**-0.866**	0.956	0.973	0.977	0.978	0.976	0.973	0.968	0.961	0.961
12	**-0.923**	0.969	0.970	0.972	0.970	0.961	0.955	0.945	0.931	0.931
13	**0.930**	0.923	0.947	0.959	0.964	0.966	0.966	0.965	0.963	0.963
14	**-0.755**	0.889	0.918	0.922	0.918	0.911	0.903	0.897	0.895	0.895
15	**-0.890**	0.936	0.951	0.957	0.961	0.963	0.964	0.963	0.961	0.961
16	**-0.835**	0.940	0.943	0.933	0.919	0.908	0.899	0.895	0.884	0.884
17	**-0.916**	0.969	0.979	0.982	0.980	0.976	0.971	0.968	0.968	0.968
18	**-0.782**	0.931	0.952	0.964	0.969	0.971	0.972	0.972	0.971	0.971
19	**-0.915**	0.951	0.973	0.980	0.979	0.975	0.966	0.952	0.940	0.940
20	**-0.729**	0.870	0.919	0.930	0.937	0.940	0.941	0.940	0.939	0.939
21	**-0.903**	0.940	0.963	0.975	0.978	0.976	0.970	0.962	0.953	0.953
22	**-0.743**	0.894	0.922	0.935	0.941	0.942	0.939	0.936	0.932	0.932
23	**-0.885**	0.923	0.934	0.938	0.945	0.951	0.956	0.958	0.954	0.954
24	**-0.882**	0.932	0.956	0.963	0.964	0.963	0.963	0.963	0.965	0.965
Median	**-0.851**	0.934	0.952	0.963	0.966	0.963	0.960	0.955	0.949	0.944

Note: Entries in boldface are below the one-tailed critical value of 0.685 commonly used for objective image quality. Alongside the PSNR, the SCG4 metric is also calculated for each of the demosaiced images.

Given these two metrics, the goal is to establish a significant correlation between them. To accomplish this, the Pearson correlation coefficient is used to analyze the degree of correlation between the two variables (PSNR and SCG4 metric). The resulting correlation coefficients are shown in Table 1. As there are 11 methods used to evaluate the correlation, there are 9 degrees of freedom (DoF) corresponding to a critical value of 0.685 for a *p*-value of 0.01. Apart from the correlation values when the significance threshold is set to 1, the metric is shown to be statistically correlated to the PSNR value. Beyond simply establishing the correlation, we also show that a threshold value of 5 results in the best median correlation for the *Kodak* database.

VI. CONCLUSION

In this work, we developed the Sparsity of Chroma Gradient in Four Directions (SCG4) metric that can be used to predict the quality of a demosaiced image obtained through a Bayer color filter array. We demonstrated certain properties of natural images particularly in relation to the chroma components and utilized these properties as a prior for our proposed metric. Experimental results show a strong correlation between SCG4 and both the subjective and objective quality of the resulting demosaiced image.

An interesting aspect of our proposed metric is its highly localized nature. Each probability estimate utilizes only 10 pixels—the central pixels for C1 and C2, and the four neighboring pixels for each of these components. This not only allows for efficient calculation but also increases the flexibility of the metric. While the smallest feasible area at which the metric can operate has yet to be investigated, it has been shown to operate with relatively small areas such as the cropped region from the Lighthouse image.

The development of such a metric, in itself, is useful in further studies of demosaicing because it can enable adaptivity in many aspects. For instance, one may develop a compound demosaicing technique that switches between various demosaicing algorithms depending on the performance on a local region in an image. More importantly, it can be used as an optimization criteria for a single demosaicing technique. Further work is being done into developing such an adaptive algorithm.

REFERENCES

[1] L. Lindgren, J. Melander, R. Johansson, and B. Moller, “A multiresolution 100-GOPS 4-Gpixels/s programmable smart vision sensor for multisense imaging,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 6, pp. 1350–1359, June 2005.

[2] T. Yamashita, R. Funatsu, T. Yanagi, K. Mitani, Y. Nojiri, and T. Yoshida, “A camera system using three 33-megapixel CMOS image sensors for UHDTV2,” *SMPTE Motion Imaging Journal*, vol. 120, no. 8, pp. 24–31, Nov 2011.

[3] J. Galt, B. Petljanski, and S. Campbell, “Development of a 70mm, 25-megapixel electronic cinematography camera with integrated flash recorder,” *SMPTE Motion Imaging Journal*, vol. 122, no. 2, pp. 42–48, March 2013.

[4] S. Kleinfelder, S. Lim, X. Liu, and A. E. Gamal, “A 10000 frames/s CMOS digital pixel sensor,” *IEEE Journal of Solid-State Circuits*, vol. 36, no. 12, pp. 2049–2059, Dec 2001.

[5] R. Berger, D. D. Rathman, B. M. Tyrrell, E. J. Kohler, M. K. Rose, R. A. Murphy, T. S. Perry, H. F. Robey, F. A. Weber, D. M. Craig, A. M. Soares, S. P. Vernon, and R. K. Reich, “A 64 64-pixel CMOS test chip for the development of large-format ultra-high-speed snapshot imagers,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 9, pp. 1940–1950, Sept 2008.

[6] R. Xu, B. Liu, and J. Yuan, “A 1500 fps highly sensitive 256 256 CMOS imaging sensor with in-pixel calibration,” *IEEE Journal of Solid-State Circuits*, vol. 47, no. 6, pp. 1408–1418, June 2012.

[7] M. Furuta, Y. Nishikawa, T. Inoue, and S. Kawahito, “A high-speed, high-sensitivity digital CMOS image sensor with a global shutter and 12-bit column-parallel cyclic A/D converters,” *IEEE Journal of Solid-State Circuits*, vol. 42, no. 4, pp. 766–774, April 2007.

[8] T. Takemoto, H. Yamashita, T. Yazaki, N. Chujo, Y. Lee, and Y. Matsuoka, “A 25-to-28 Gb/s high-sensitivity (-9.7 dBm) 65 nm CMOS optical receiver for board-to-board interconnects,” *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2259–2276, Oct 2014.

[9] B. E. Bayer, “Color imaging array,” July 1976, US Patent 3,971,065.

[10] “Kodak lossless true color image suite,” 2013, Retrieved 2016-11. [Online]. Available: http://r0k.us/graphics/kodak/
[11] 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, no. 1, pp. 44–54, Jan 2005.

[12] B. Leung, G. Jeon, and E. Dubois, “Least-squares lumachroma demultiplexing algorithm for bayer demosaicking,” *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1885–1894, July 2011.

[13] R. Kimmel, “Demosaicing: image reconstruction from color CCD samples,” *IEEE Transactions on Image Processing*, vol. 8, no. 9, pp. 1221–1228, Sep 1999.

[14] J. E. Adams, Jr., “Interactions between color plane interpolation and other image processing functions in electronic photography,” *Proc. SPIE*, vol. 2416, pp. 144–151, 1995. [Online]. Available: http://dx.doi.org/10.1117/12.204825

[15] S.-C. Pei and I.-K. Tam, "Effective color interpolation in CCD color filter array using signal correlation," in *Proceedings 2000 International Conference on Image Processing* (Cat. No.00CH37101), vol. 3, 2000, pp. 488–491 vol.3.

[16] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 997–1013, Sep 2002.

[17] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Beyond color difference: Residual interpolation for color image demosaicking," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1288–1300, March 2016.

[18] T. Yu, W. Hu, W. Xue, and W. Zhang, "Colour image demosaicking via joint intra and inter channel information," Electronics Letters, vol. 52, no. 8, pp. 605–607, 2016. [19] K. Hirakawa and T. W. Parks, "Adaptive homogeneitydirected demosaicing algorithm," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 360–369, March 2005.

[20] P. Getreuer, "Gunturk–Altunbasak–Mersereau alternating projections image demosaicking," *Image Processing On Line*, 2011.

[21] ——, "Color demosaicing with contour stencils," in 2011 *17th International Conference on Digital Signal Processing (DSP)*, July 2011, pp. 1–6.

[22] ——, "Image demosaicking with contour stencils," *Image Processing On Line*, vol. 2, pp. 22–34, 2012.

[23] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2167–2178, Dec 2005.

[24] P. Getreuer, "Zhang-Wu directional LMMSE image demosaicking," *Image Processing On Line*, 2011.

[25] H. S. Malvar, L. wei He, and R. Cutler, "High-quality linear interpolation for demosaicing of bayer-patterned color images," in 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 3, May 2004, pp. iii–485–8 vol.3.

[26] P. Getreuer, "Malvar-He-Cutler linear image demosaicking," *Image Processing On Line*, vol. 1, 2011.

[27] X. Li, "Demosaicing by successive approximation," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 370–379, March 2005.

Development and Implementation of an Integrated Services Digital Broadcasting-Terrestrial (ISDB-T) Decoder Using GNU Radio and USRP1

Alberto S. Bañacia, Joseph C. Acebedo, Johari C. Villegas, and Arvin Tate N. Yu

Abstract—**The recent adoption by the Philippine government of Japan's Integrated Services Digital Broadcasting- Terrestrial (ISDB-T) standard as the digital TV standard to be used in the country prompts the need for a platform that will facilitate understanding of its concept and provide a framework for evaluating algorithms that will improve ISDB-T receiver's performance. This research work develops and implements an ISDB-T decoder in a Software-Defined Radio (SDR) platform. First, a recorded raw baseband ISDB-T signal in .dat file format was simulated through the configuration of signal processing blocks in GNU Radio Companion (GRC). The successful simulation produced a transport stream H264-MPEG-4 AVC file and resulted to the display of the signal in a 6MHz bandwidth with a 64QAM constellation plot, a 34.22 dB Modulation Error Rate (MER), 8.95 dB and 18.00 dB Bit Error Rate (BER) for the Viterbi and Reed Solomon decoders, respectively. The simulation measurements served as indicators for the quality of the decoded signal as the output transport stream file was played simultaneously in VLC media player. Second, an actual reception of a private Philippine ABS-CBN Digital Television Terrestrial (DTT) signal located in UHF Channel 43 was performed in real-time through the system integration of the USRP1 with the GRC Signal flow graph. A Log Periodic LP0410 PCB antenna was also integrated to the SDR platform to receive the digital broadcast. Due to proprietary DTV issues, the system was only able to display ABS-CBN's frequency spectrum centered at its assigned frequency of 611.143MHz at a 6MHz bandwidth.**

Index terms—**digital video broadcasting, ISDB-T, software radio, GNU Radio, USRP1**

I. Introduction

SINCE the 1940s, the broadcast industry in the Philippines has been operating in analog terrestrial broadcast. One setback of analog broadcasting, aside from being susceptible to noise and of low quality, is its excessive use of frequency spectrum. Frequency spectrum is a limited resource and has to be conserved. Over time in parallel with current advancements of technology, the world is now undergoing a transition from analog terrestrial broadcast to digital terrestrial broadcast in order to address the problem of perceived spectrum scarcity. There has been a number of existing standards in DTT adopted by different countries including Europe's Digital Video Broadcasting-Terrestrial (DVB-T), North America's Advanced Television System Committee (ATSC), China's Digital Terrestrial/ Television Multimedia Broadcasting (DTMB) and Japan's ISDB-T. In 2010, the Philippine government, through the National Telecommunications Commission (NTC), has officially announced the adoption of Japan's ISDB-T citing as one of its main reasons for adoption its distinct feature of having an Early Warning Broadcasting Services (EWBS) that could be an effective tool in disasters and calamity preparedness and mitigation. Just like DVB-T, ISDB-T also adopts Orthogonal Frequency Division Multiplexing (OFDM).

Being a relatively new technology in the Philippine setting, a lot of things still are to be learned and understood about ISDB-T standard. Digital TV broadcast using ISDB-T transmission in the country is still in an experimental stage and complete migration from analog to digital TV is not expected until year 2020. Hence, an

Alberto S. Bañacia, Joseph C. Acebedo, Johari C. Villegas, and Arvin Tate N. Yu are with the Department of Electrical and Electronics Engineering, University of San Carlos, Cebu City, Philippines
abanacia1201@gmail.com,
arvintateyu@gmail.com

ISDB-T receiver platform that is reconfigurable, flexible, and programmable is necessary to facilitate understanding of its concept and provide a framework for evaluating algorithms that will improve receiver's performance.

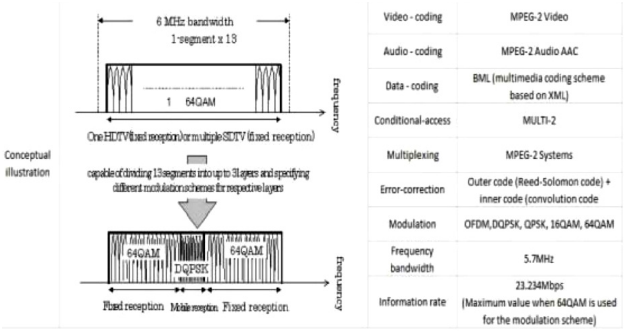
In this paper, an SDR platform was used to implement an ISDB-T decoder. The GRC software combined with the Universal Software Radio Peripheral (USRP) hardware forms the platform. The implementation was initially done by the simulation of the created grc flow graph in GNU Radio that is installed in a PC running on Ubuntu OS. To receive an over-the-air ISDB-T signal, the PC is USB-interfaced with the USRP1 hardware connected to a log periodic antenna. The output was then displayed in the host PC through the use of the VLC media player.

The rest of this paper is organized as follows. In section II, the basics of the ISDB-T standard and SDR are discussed. This is followed by section III that describes the implementation of an ISDB-T decoder in an SDR platform. Section IV presents and discusses the results of the study. The paper is concluded in section V while section VI expresses the authors' acknowledgement.

II. Basics of ISDB-T System and Software-defined Radio

A. *ISDB-T Standard*

ISDB-T system is a Japanese standard that is intended to deliver high quality data, video and sound for Digital Terrestrial Television Broadcasting (DTTB). The standard, which was developed in Japan, utilizes Band Segmented Transmission - Orthogonal Frequency Division Multiplexing (BSTOFDM). This method of multiplexing is flexible since a set of OFDM segments having different parameters for transmission is being used. Also hierarchical modulation is permissible in ISDB-T. The system has an Early Warning System (EWS) that helps in the information dissemination in the event of natural disasters such as that of a typhoon, tsunami or earthquake.


6 MHz bandwidth
1-segment x 13
1 64QAM
frequency
One HDTV (fixed reception) or multiple SDTV (fixed reception)
Conceptual illustration
capable of dividing 13 segments into up to 3 layers and specifying different modulation schemes for respective layers
64QAM
DQPSK
64QAM
frequency
Fixed reception
Mobile reception
Fixed reception

Video - coding	MPEG-2 Video
Audio - coding	MPEG-2 Audio AAC
Data - coding	BML (multimedia coding scheme based on XML)
Conditional-access	MULTI-2
Multiplexing	MPEG-2 Systems
Error-correction	Outer code (Reed-Solomon code) + inner code (convolution code
Modulation	OFDM,DQPSK, QPSK, 16QAM, 64QAM
Frequency bandwidth	5.7MHz
Information rate	23.234Mbps (Maximum value when 64QAM is used for the modulation scheme)

Fig. 1. Overview of the digital terrestrial television broadcast system [5]

Figure 1 presents the overview for one channel of the ISDB-T system. One channel is split into 13 segments. 12 of 13 segments are used for Digital TV transmission which uses MPEG-2 Video coding and MPEG-2 Audio AAC coding. Broadcast Markup Language (BML), which is a multimedia coding scheme based on XML (Extensible Markup Language), is used for Data coding. ISDB-T adopts MULTI-2 Conditional Access which is used to scramble digital multimedia content. While receiving noise and interference, ISDB-T signals go through error correction. There are two types of error correcting codes being used namely Reed-Solomon code (Outer code) and Convolutional Coding (Inner code). ISDB-T also has different modulation schemes: DQPSK, QPSK, 16QAM and 64QAM. In one channel, ISDB-T signals have a bandwidth of 5.7 MHz, close to 6MHz, and if 64QAM is being used as a modulation scheme, ISDB-T signal will have a maximum information rate of about 23.234 Mbps (Megabits per second). 1 out of 13 signals is transmitted for mobile reception

a) ISDB-T Transmission System Configuration

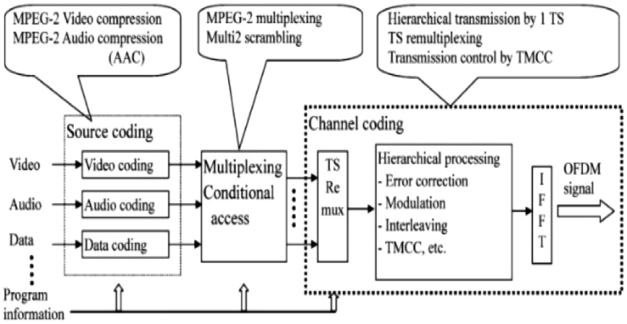


Fig. 2. Configurations for ISDB-T System [7]

As shown in Figure 2, video, audio, and some other service data signals that are being encoded are multiplexed to form elementary streams. These elementary streams are multiplexed again with other additional programs to form the MPEG-2 compression Transport Stream (TS). This transport stream goes through hierarchical processing like error correction, modulation, and interleaving. ISDB-T signals also make use of TMCC (Transmission Multiplexing Configuration Control) where transmission control information such as channel segment configuration, transmission parameters, etc. are sent to the receiver. Then Inverse Fast Fourier Transform (IFFT) is performed to produce an OFDM signal.

Table 1 shows the parameters for transmission of the ISDB-T system. It utilizes 13 segments with a transmission bandwidth of 5.575MHz, 5.573MHz, and 5.572MHz for the different modes. The guard-interval ratio, inner and outer coding rate and information bit rate have different values depending on the modulation scheme selected. Out of these three modes, Mode 3 is being adopted here in the Philippines.

TABLE I

ISDB-T Transmission Parameters [7]

Transmission Parameter	Mode 1	Mode 2	Mode 3
No. of OFDM segments	13		
Bandwidth	5.575MHz	5.573MHz	5.572MHz
Carrier interval	3.968kHz	1.984kHz	0.992kHz
No. of carriers	1405	2809	5617
Carrier modulation	QPSK,16QAM,64QAM,DQPSK		
Effective symbol length (Tu)	252μs	504μs	1.008ms
Guard interval length (Tg)	1/4, 1/8, 1/16, 1/32 of effective symbol length		
No. of symbols per frame	w 204		
Time interleave	Maximum 4 values: 0,0.1,0.2,0.4 sec		
Frequency interleave	Intra-segment and inter-segment interleaving		
Inner code	Convolutional coding (1/2, 2/3,3/4,5/6, 7/8)		
Outer code	RS (204, 188)		
Information bit rate	3.65Mbps – 23.23Mbps		
Heirarchial transmission	Maximum 3 levels (Layer A, B and C)		

Among the technical advantages of the ISDB-T standard are [8]:

- High quality and service flexibility
- Robustness and reception flexibility
- Effective utilization of frequency resource
- One-seg service
- Commonality
- Utilization for disaster prevention

b) ISDB-T Decoder Error Rates

- Modulation Error Rate (MER)

MER is computed when there is error present in the receiver that would degrade the system. It indicates if the signal is indeed decoded correctly by the receiver. MER is calculated using Equation (1):

$$MER = 10 \times \log 10 \left\{ \frac{\sum_{j=1}^{N} \left(I^2{}_j + Q^2 \right)}{\sum_{j=1}^{N} \left(\delta I^2{}_j + \delta Q^2 \right)} \right\} \quad (1)$$

Where I_j and Q_j are vector points of the constellation diagram and δI_j and δQ_j are the error vectors. We can determine the actual received vector points $\tilde{I}_j$ and $\tilde{Q}_j$ by getting the sum of the ideal vector points and the error vectors. Therefore:

$$\tilde{I}_j = I_j + \delta I_j \text{ and } \tilde{Q}_j = Q_j + \delta Q_j \quad (2)$$

- Bit Error Rate (BER)

BER is computed as the number of error bits that are present out of the total number of bits in percentage. Ideally, having a very low BER will result to a better reception of the signal since with lower percentage, there are fewer errors indicating that the signal can easily be corrected and properly decoded. *BER* is determined by the general equation:

$$BER = \frac{B_E}{B_T} \times 100\% \quad (3)$$

where B_E is the number of error bits that have been received and B_T is the number of total bits that have been received. Reed Solomon codes are usually in the form RS(n,k) with *s*-bit symbols. We denote *n* to be the code word length of the Reed Solomon codes and *k* to be the data symbols of *s* bits and *P* denotes the parity bits. Therefore, in equation form:

$$n = 2^s - 1 \quad \text{and} \quad n = k + P \quad (4)$$

Taking into account that a Reed Solomon decoder can correct up to *t* symbols that contains errors, equation (4) can also be rewritten as:

$$n = 2^s - 1 \quad \text{and} \quad n = k + 2t \quad (5)$$

B. Software-Defined Radio

SDR has been the trend in wireless communications. SDR has a unique function where radios are implemented in software instead of being implemented completely in hardware. It means that hardware functionalities and processes can be implemented in software domain.

SDR is highly favored in wireless systems since it is known for its flexibility. Modifications in systems such as protocols and algorithms can be done easily since they are reprogrammable, reconfigurable and capable of supporting different and diverse radio and networking systems [9]. Against Analog radio, SDR has relatively low maintenance and operation costs, has a wide variety of alternate inexpensive solutions and capable of operating in different signal bands. Elements like filters, mixers and modulators which are in practice are analog elements, can be implemented digital form trough software.

C. GNU Radio

GNU Radio is a free, open source software that is used to develop software radios through their signal processing blocks. GNU Radio applications are normally coded in Phyton language and supported by C++ language.

Among the elements or blocks contained in the GNU Radio are filter blocks, channel code, synchronization element blocks, equalizer, demodulator, decoder, and other components which are normally found in a real-world radio systems but traditionally implemented in hardware.

Once these blocks are properly connected, these can be used to simulate real-world radio communication systems to allow performance analysis even prior to actual implementations so that appropriate modifications could be performed on certain algorithms or functionalities[9].

III. SDR Implementation of an ISDB-T Decoder

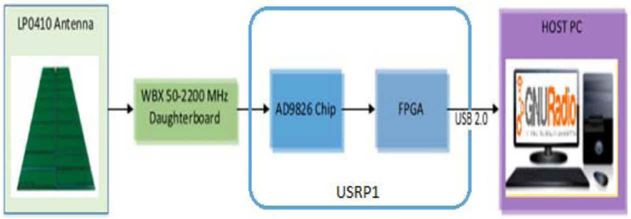


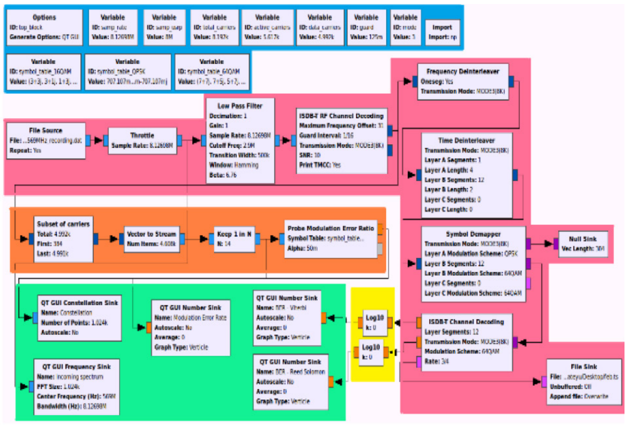
Fig. 3. System Architecture in Software-Defined Radio Platform

Shown in Figure 3 is the ISDB-T decoder in an SDR Platform. The ISDB-T signal to be decoded was received by a LP0410 Antenna which operates in the frequency range of 400 MHz to 1GHz with a 5 to 6dBi gain. The antenna was connected to the receiver port of a WBX daughterboard. The WBX daughterboard which operates in the 50–2200 MHz frequency range was used for this project and was connected to one of the four slots of the motherboard as an RF receiver interface or tuner. The WBX offers a bandwidth capacity of 40MHz and is more than enough to satisfy the 6 – MHz band requirement of the ISDB-T signal. The daughterboard was connected to the USRP1 hardware motherboard. The host PC is installed with an Ubuntu Linux-based operating system.

A. Software Simulation

GNU Radio applications are primarily written using the Python programming language (the GNU Radio Companion is actually a front-end that automatically generates Python code). However, most blocks are written internally in C++: in this sense, Python may be regarded as the scripting language used by GNU Radio. Thus, real-time, high-throughput radio systems can be developed using GNU Radio.

A1. Raw Baseband ISDB-T Signal Simulation



Options
ID: top_block
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 8.12698M

Variable
ID: samp_usrp
Value: 8M

Variable
ID: total_carriers
Value: 8.192k

Variable
ID: active_carriers
Value: 5.617k

Variable
ID: data_carriers
Value: 4.992k

Variable
ID: guard
Value: 125m

Variable
ID: mode
Value: 3

Import
Import: np

Variable
ID: symbol_table_16QAM
Value: (3+3j, 3+1j, 1+3j, ...

Variable
ID: symbol_table_QPSK
Value: 707.107m...m-707.107mj

Variable
ID: symbol_table_64QAM
Value: (7+7j, 7+5j, 5+7j, ...

File Source
File: ...569MHz_recording.dat
Repeat: Yes

Throttle
Sample Rate: 8.12698M

Low Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 8.12698M
Cutoff Freq: 2.9M
Transition Width: 500k
Window: Hamming
Beta: 6.76

ISDB-T RF Channel Decoding
Maximum Frequency Offset: 31
Guard Interval: 1/16
Transmission Mode: MODE3(8K)
SNR: 10
Print TMCC: Yes

Frequency Deinterleaver
Oneseg: Yes
Transmission Mode: MODE3(8K)

Time Deinterleaver
Transmission Mode: MODE3(8K)
Layer A Segments: 1
Layer A Length: 4
Layer B Segments: 12
Layer B Length: 2
Layer C Segments: 0
Layer C Length: 0

Subset of carriers
Total: 4.992k
First: 384
Last: 4.991k

Vector to Stream
Num Items: 4.608k

Keep 1 in N
N: 14

Probe Modulation Error Ratio
Symbol Table: symbol_table...
Alpha: 50m

Symbol Demapper
Transmission Mode: MODE3(8K)
Layer A Modulation Scheme: QPSK
Layer B Segments: 12
Layer B Modulation Scheme: 64QAM
Layer C Segments: 0
Layer C Modulation Scheme: 64QAM

Null Sink
Vec Length: 384

QT GUI Constellation Sink
Name: Constellation
Number of Points: 1.024k
Autoscale: No

QT GUI Number Sink
Name: Modulation Error Rate
Autoscale: No
Average: 0
Graph Type: Verticle

QT GUI Number Sink
Name: BER - Viterbi
Autoscale: No
Average: 0
Graph Type: Verticle

Log10
k: 0

Log10
k: 0

ISDB-T Channel Decoding
Layer Segments: 12
Transmission Mode: MODE3(8K)
Modulation Scheme: 64QAM
Rate: 3/4

QT GUI Frequency Sink
Name: incoming spectrum
FFT Size: 1.024k
Center Frequency (Hz): 569M
Bandwidth (Hz): 8.12698M

QT GUI Number Sink
Name: BER - Reed Solomon
Autoscale: No
Average: 0
Graph Type: Verticle

File Sink
File: ...ateyu/Desktop/feb.ts
Unbuffered: Off
Append file: Overwrite

Fig. 4. Raw Baseband ISDB-T signal GRC flowgraph

Shown in Figure 4 is the flowgraph of the ISDB-T decoder capable of simulating the raw baseband signal. In this flowgraph, it can display both the decoded ISDB-T video and the measurement for the MER, BER Viterbi and BER Reed Solomon codes. Highlighted in blue are the Variable blocks, Option block and Import block in utility to the configuration of the main blocks. The main blocks (highlighted in pink) are responsible in demodulating the "569MHz_recording.dat" file which consists of the File Source block, Throttle block, Low Pass Filter block, ISDBT-RF Channel Decoding block, Frequency Deinterleaver block, Time Deinterleaver block, Symbol Demapper, Null Sink, ISDB-T Channel Decoding block and File Sink block. The purpose of the blocks highlighted in orange is to compute the MER and to determine the modulation scheme. The Log10 block (highlighted in yellow) converts the BER ratio to decibels. Highlighted in green are the blocks responsible for the display of the following: the QT GUI Frequency Sink block for the frequency spectrum, the QT GUI Constellation Sink block for the the constellation plot and the QT GUI Number Sink block for the measurements of the MER, BER-Viterbi, and BER-Reed Solomon.

The first stage of the decoding process is selecting the 569MHz_recording.dat file as an input for the File Source block. A throttle is used to limit the CPU usage while the flowgraph is being generated. It is sampled at a frequency of 512/63 ≈ 8.12698MHz which is the frequency used to receive an OFDM signal. A Low Pass Filter block with a cut-off frequency fc = 2.9MHz is used to filter neighboring channels to avoid noise interference in the system. What follows is the ISDB-T RF Channel Decoding which performs the decoding depending on the mode of the hierarchical layer being used, its guard interval, and SNR value. In this case: The transmission mode being used is MODE3(8K) and the guard interval is 1/16 of the active symbol duration. In 8K mode, the active symbol duration is 1.008ms. The SNR value is set to 10. Finally, the ISDB-T RF Channel Decoding block will detect the end of every OFDM frame (the grouping of 204 OFDM symbols that transmit a complete TMCC) and perform the TMCC decoding.

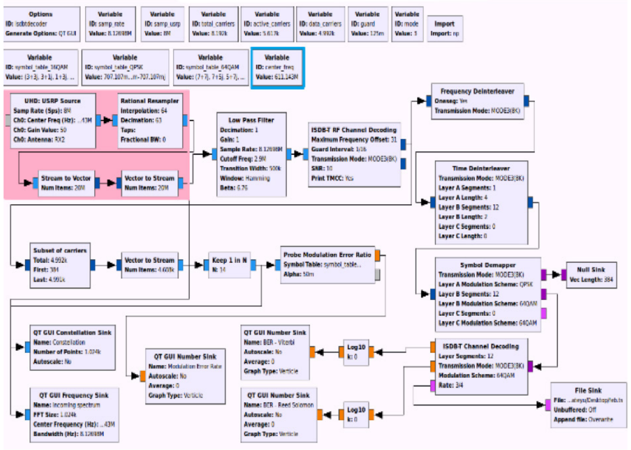
For those received data carriers, frequency and time de-interleaving are performed and then symbol de-mapping followed by bit de-interleaving. 64QAM and 16QAM are the modulation schemes usually used for full-segment receivers while QPSK is used for handheld one-segment receivers. The Symbol De-mapper is used to determine the different modulation schemes of each hierarchical layer. Layer A uses one out of 13 segments of the 6MHz bandwidth with a modulation scheme of QPSK. Layer B uses 12 out of 13 segments of the 6MHz bandwidth with a modulation scheme of 64QAM. Layer C is unused. The ISDB-T Channel Decoding decodes the channel from the 12 segments of the 6MHz bandwidth. The channel is decoded depending on the transmission mode (MODE1 2K, MODE2 4K, MODE3 8K), modulation scheme (QPSK, 16QAM, 64QAM), and convolutional rate (1/2, 2/3, ¾, 5/6, 7/8). In this case: The 12 layer segments are being decoded by using MODE3 as the transmission mode with a modulation scheme of 64QAM, and a convolutional rate of ¾. The ISDB-T Channel Decoding decodes the channel from the 12 segments of the 6MHz bandwidth. The channel is decoded depending on the transmission mode (MODE1 2K, MODE2 4K, MODE3 8K), modulation scheme (QPSK, 16QAM, 64QAM), and convolutional rate (1/2, 2/3, ¾, 5/6, 7/8). In this case, the 12 layer segments are being decoded by using MODE3 as the transmission mode with a modulation scheme of 64QAM and a convolutional rate of ¾.

The basic idea of having Reed Solomon code is that it checks each byte for each link in the transmission chain. RS codes correct all errors and produce an error-free TS packet. Under normal operational circumstances, the RS decoder will correct all errors and produce an error-free TS packet.

The Viterbi decoder reconstructs the original streams of data that entered a convolutional encoder. This would be possible with the maximum likelihood decoding which bases its error rate from a correct statistical model and checking if the Viterbi decoder reconstructed the streams of data.

Lastly, the File Sink block is used to create a TS file format which contains the decoded signal from the File Source block. The TS file will then be played in VLC Media Player to test whether the signal was properly received or not. A successful reception depends on the quality of the signal being received that includes noise, inter-modulation, and other corruptions of the signal.

A2. Real-Time Reception of Local ISDB-T Signal



Options
ID: isdbtdecoder
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 8.12698M

Variable
ID: samp_usrp
Value: 8M

Variable
ID: total_carriers
Value: 8.192k

Variable
ID: active_carriers
Value: 5.617k

Variable
ID: data_carriers
Value: 4.992k

Variable
ID: guard
Value: 125m

Variable
ID: mode
Value: 3

Import
Import: np

Variable
ID: symbol_table_16QAM
Value: (3+3j, 3+1j, 1+3j, ...

Variable
ID: symbol_table_QPSK
Value: 707.107m...m-707.107mj

Variable
ID: symbol_table_64QAM
Value: (7+7j, 7+5j, 5+7j, ...

Variable
ID: center_freq
Value: 611.143M

UHD: USRP Source
Samp Rate (Sps): 8M
Ch0: Center Freq (Hz): ...43M
Ch0: Gain Value: 50
Ch0: Antenna: RX2

Rational Resampler
Interpolation: 64
Decimation: 63
Taps:
Fractional BW: 0

Stream to Vector
Num Items: 20M

Vector to Stream
Num Items: 20M

Low Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 8.12698M
Cutoff Freq: 2.9M
Transition Width: 500k
Window: Hamming
Beta: 6.76

ISDB-T RF Channel Decoding
Maximum Frequency Offset: 31
Guard Interval: 1/16
Transmission Mode: MODE3(8K)
SNR: 10
Print TMCC: Yes

Frequency Deinterleaver
Oneseg: Yes
Transmission Mode: MODE3(8K)

Time Deinterleaver
Transmission Mode: MODE3(8K)
Layer A Segments: 1
Layer A Length: 4
Layer B Segments: 12
Layer B Length: 2
Layer C Segments: 0
Layer C Length: 0

Subset of carriers
Total: 4.992k
First: 384
Last: 4.991k

Vector to Stream
Num Items: 4.608k

Keep 1 in N
N: 14

Probe Modulation Error Ratio
Symbol Table: symbol_table...
Alpha: 50m

Symbol Demapper
Transmission Mode: MODE3(8K)
Layer A Modulation Scheme: QPSK
Layer B Segments: 12
Layer B Modulation Scheme: 64QAM
Layer C Segments: 0
Layer C Modulation Scheme: 64QAM

Null Sink
Vec Length: 384

QT GUI Constellation Sink
Name: Constellation
Number of Points: 1.024k
Autoscale: No

QT GUI Number Sink
Name: Modulation Error Rate
Autoscale: No
Average: 0
Graph Type: Verticle

QT GUI Number Sink
Name: BER - Viterbi
Autoscale: No
Average: 0
Graph Type: Verticle

Log10
k: 0

ISDB-T Channel Decoding
Layer Segments: 12
Transmission Mode: MODE3(8K)
Modulation Scheme: 64QAM
Rate: 3/4

QT GUI Frequency Sink
Name: Incoming spectrum
FFT Size: 1.024k
Center Frequency (Hz): ...43M
Bandwidth (Hz): 8.12698M

QT GUI Number Sink
Name: BER - Reed Solomon
Autoscale: No
Average: 0
Graph Type: Verticle

Log10
k: 0

File Sink
File: ...ateyu/Desktop/feb.ts
Unbuffered: Off
Append file: Overwrite

Fig. 5. Reception of Local ISDB-T signal GRC flowgraph

Figure 5 shows the GRC flowgraph for the reception of ABS-CBN ISDB-T signal. Highlighted in blue is the center_freq variable block set to a value of 611.143MHz which is the center frequency of ABS-CBN. The flowgraph process is similar to Figure 4 but this time the initial stage, which is highlighted in pink, is composed of the UHD: USRP Source block, Rational Resampler block, Streams to Vector block, and Vector to Streams block. The broadcasted signals from the ABS-CBN transmitter was received by the LP0410 antenna then fed to the USRP1. The WBX daughterboard in the USRP hardware performs RF to IF signal conversion and the motherboard converts the IF signal to baseband signal and sends the signal into the PC via USB 2.0 connectivity, which is to be simulated by the GRC blocks. The UHD: USRP Source block reads the received signal and samples them with a frequency of 8MHz. As these devices usually cannot sample at any arbitrary rate, a rational re-sampler is used to obtain a sampling rate of 512/63 ≈ 8.12698MHz. To obtain the ratio of 512/63 MHz, the sampling rate of the USRP1, which is 8MHz, is then multiplied by the interpolation value of 64 and divided by the decimation value of 63. The Streams to Vector block connected to Vector to Streams block acts as a sort of buffer for the signal to minimize dropping samples of the decoded signal. Then, the rest of the blocks in the flowgraph is generated in the same way as the previous figure.

B. Decoder Implementation with GRC and USRP

USRP1, the original hardware of the USRP™, was interfaced with GNU radio. The hardware motherboard is composed of a 64 MS/s dual ADC, 128 MS/s dual DAC and an Altera Cyclone FPGA with high-speed USB 2.0 that was connected to the host PC. The FPGA is responsible for high bandwidth computations and the decrease of data rate for USB 2.0 transmission while two Analog Device (AD9826) chips handles the conversion of analog and digital signals. The AD9826 chip has a 16-bit A/D converter that converts the analog signal being received by the USRP into a digital signal.

An ISDB-T signal was decoded through the use of Signal Processing blocks in the GNU Radio Companion software tool. The UHD: USRP Source block processes the ISDB-T signals from the TV Broadcasting station which is received by the antenna. Also, a simulation of a raw baseband ISDB-T signal was done by using the File source block. The file source block allows a recorded video .dat file (a raw baseband samples) as the file to be processed. Once the receiver acquired an ISDB-T signals from the transmitter, it underwent several processing including its conversion from RF to IF, Analog to Digital, filtering, Channel decoding, Time and Frequency interleave, RS coding, Viterbi. After the digital signal processing has been done, the data will become a transport stream. By using File Sink block, the .ts file output is compiled, thus bytes of data are stored. The .ts file can be viewed using VLC media player. A successful reception will result to a clear decoded ISDB-T signals.

IV. Results and Discussions

A. Simulation of raw baseband ISDB-T Signals

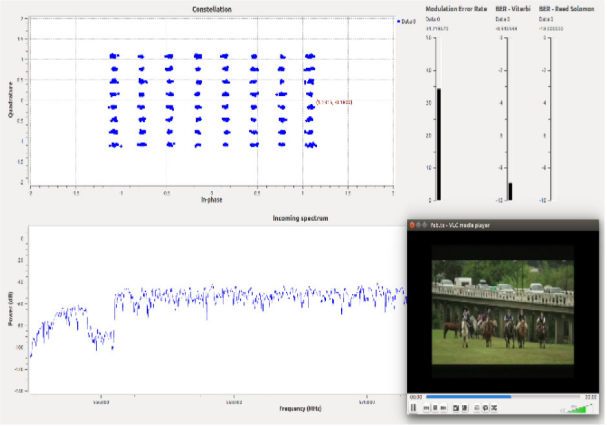


Fig. 6. Simulation using recorded .dat file

Figure 6 shows the constellation diagram and measurements of BER and MER. On the rightmost side is the .ts file output which is played simultaneously while running the ISDB-T receiver system. The .ts file is stored and can be viewed in VLC media player to display a high-quality video with clear audio while the system is decoding the raw baseband signal.

(1) ISDB-T signal spectrum

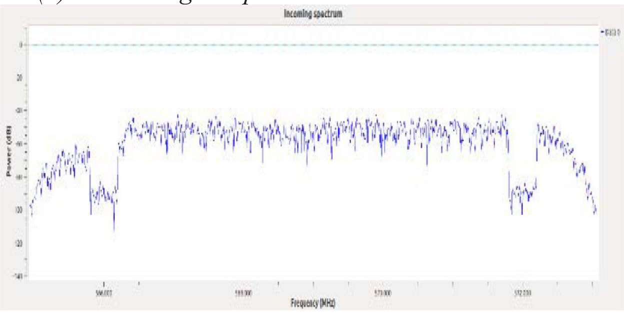
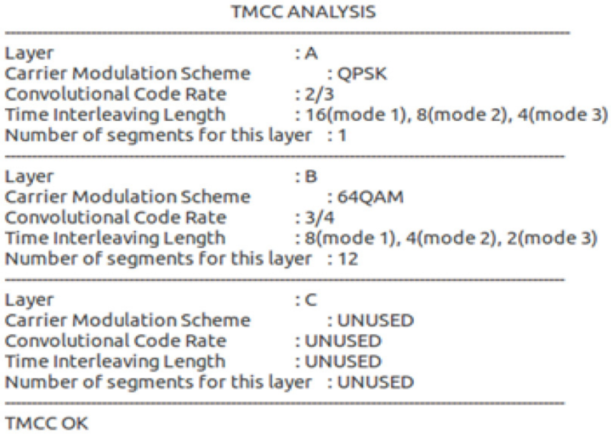


Fig. 7. Raw Baseband ISDB-T Signal Spectrum

The signal spectrum is used to determine if the signal is received by the hardware. An ISDB-T signal with a bandwidth of 6MHz is shown in Figure 7, it is the displayed signal spectrum in the QT GUI Frequency Sink block of GRC. It is shown that the bandwidth displayed is within range of values 566.000MHz to 572.000MHz which would give a 6MHz value. This displayed bandwidth shows that an OFDM signal is present and that an ISDB-T signal is received properly.

(2) TMCC

It is shown in the TMCC analysis (Figure 8) of the raw baseband samples that the signal was properly decoded and corrected. Layer A was used for the 1-seg receiver since there is only one segment length of the OFDM bandwidth. The convolutional rate of Layer A is shown to have a value of 2/3 and a time interleaving length of 16 in MODE1(2K), 8 in MODE2(4K), and 4 in MODE3(8K). In this case, it has a length of 4 since MODE3(8K) was used. Layer B used 12 out of the 13 segments of the OFDM bandwidth. It used 64QAM as the modulation scheme, had a convolutional rate of 3/4, and a time interleaving length of 8 in MODE1(2K), 4 in MODE2(4K), and 2 in MODE3(8K).



```
                         TMCC ANALYSIS
------------------------------------------------------------------
Layer                            : A
Carrier Modulation Scheme              : QPSK
Convolutional Code Rate          : 2/3
Time Interleaving Length         : 16(mode 1), 8(mode 2), 4(mode 3)
Number of segments for this layer  : 1
------------------------------------------------------------------
Layer                            : B
Carrier Modulation Scheme              : 64QAM
Convolutional Code Rate          : 3/4
Time Interleaving Length         : 8(mode 1), 4(mode 2), 2(mode 3)
Number of segments for this layer  : 12
------------------------------------------------------------------
Layer                            : C
Carrier Modulation Scheme              : UNUSED
Convolutional Code Rate          : UNUSED
Time Interleaving Length         : UNUSED
Number of segments for this layer  : UNUSED
------------------------------------------------------------------
TMCC OK
```

Fig. 8. TMCC Analysis of the Raw Baseband ISDB-T signal

In this case, it has a length of 2 since MODE3(8K) was used. Layer C is not used since it can be used as an alternative option for the 12 segments. In this case, it is unused since all OFDM segments have been used up.

(3) Constellation diagram and MER

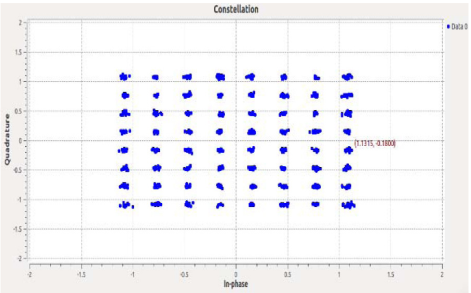


Fig. 9. Constellation Diagram of the Raw Baseband ISDB-T signal

Figure 9 shows the result of the 569MHz_recording.dat file with a modulation scheme of 64QAM. Notice that the constellation points were properly placed with minimal errors. This means that the raw baseband samples have been correctly demodulated.

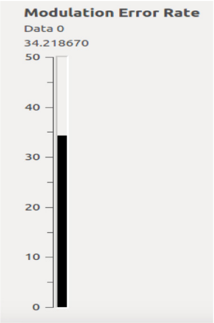


Fig. 10. Modulation Error Rate of the Raw Baseband ISDB-T signal

Figure 10 shows that MER of the decoded video is around a value of 34.22dB which is greater than that of 20 dB in [3]. This is an implication that the signal is properly modulated with a 64QAM modulation scheme.

4) Bit error rate

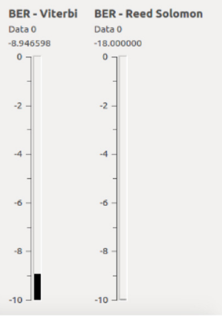


Fig. 11. BER of the Raw Baseband ISDB-T Signal (Convolutional Rate = 3/4)

The data in figure 11 is shown to have a value of 18.00dB in the BER-Reed Solomon measurement and around 8.95dB in the BER-Viterbi measurement. In the ISDB-T full-seg receiver GRC file, the ISDB-T Channel Decoding block had a convolutional rate of 3/4. This is somewhat ideal since to properly decode a signal, the BER of the signal must be of a smaller value. These results show that the raw baseband samples were properly decoded without much errors since the BER value is smaller.

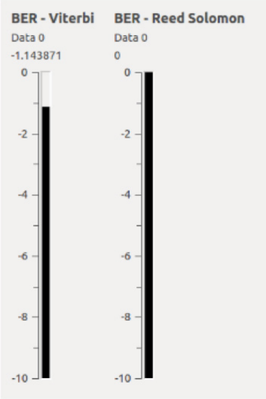
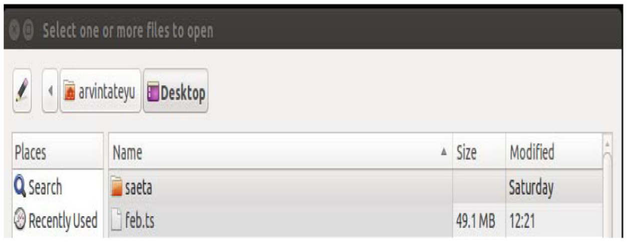


Fig. 12. BER of the Raw Baseband ISDB-T Signal (Convolutional Rate ≠ 3/4)

When the convolutional rate was changed, it is shown in Figure 12 that the BER-Viterbi decoder measurement obtained a smaller value of 1.143dB and the BER-Reed Solomon decoder obtained a value of 0dB. This indicates that the signal received is not decodeded properly.

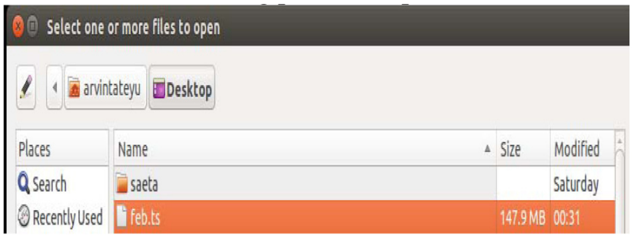
(5) Transport stream (TS) Analysis



Name	Size	Modified
saeta		Saturday
feb.ts	49.1 MB	12:21

Fig. 13. Simulated Raw Baseband Samples (No Repeat)

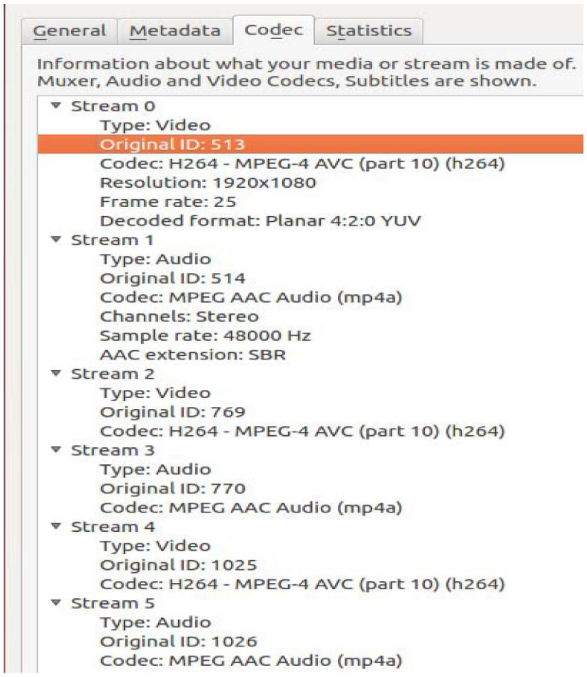
The output transport stream file set in the File Sink block was produced upon simulation of the decoder. Shown in Figure 13 is a .ts file named feb.ts which had an increase in file size as the decoder is being simulated. Without enabling the repeat option in the File Source block, the decoded raw baseband samples had a maximum value of 49.1MB which only display a video with a 21-second duration. After the .ts file reaches this value, the raw baseband signal is no longer simulated and the decoding process stops.



```
Select one or more files to open
arvintateyu   Desktop
Places          Name                Size      Modified
Search          saeta                         Saturday
Recently Used   feb.ts              147.9 MB  00:31
```

Figure 14. Simulated Raw Baseband Samples (With Repeat)

Shown in Figure 14 is the same .ts file named feb.ts, but this time, the repeat option in the File Source block was enabled. Given this, the GRC blocks continue to decode the raw baseband samples until it reaches its maximum file size, which is 1.5GB, and after the .ts file reaches this value, the raw baseband signal is no longer decoded and the process stops.

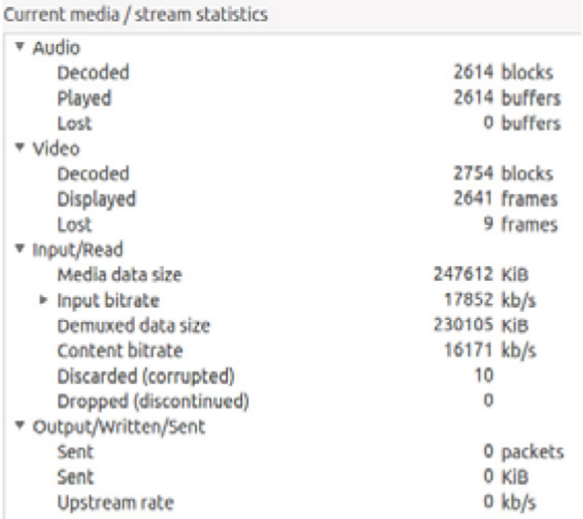


```
General   Metadata   Codec   Statistics

Information about what your media or stream is made of.
Muxer, Audio and Video Codecs, Subtitles are shown.
▾ Stream 0
    Type: Video
    Original ID: 513
    Codec: H264 - MPEG-4 AVC (part 10) (h264)
    Resolution: 1920x1080
    Frame rate: 25
    Decoded format: Planar 4:2:0 YUV
▾ Stream 1
    Type: Audio
    Original ID: 514
    Codec: MPEG AAC Audio (mp4a)
    Channels: Stereo
    Sample rate: 48000 Hz
    AAC extension: SBR
▾ Stream 2
    Type: Video
    Original ID: 769
    Codec: H264 - MPEG-4 AVC (part 10) (h264)
▾ Stream 3
    Type: Audio
    Original ID: 770
    Codec: MPEG AAC Audio (mp4a)
▾ Stream 4
    Type: Video
    Original ID: 1025
    Codec: H264 - MPEG-4 AVC (part 10) (h264)
▾ Stream 5
    Type: Audio
    Original ID: 1026
    Codec: MPEG AAC Audio (mp4a)
```

Fig. 15. CODEC Information of the TS File

Figure 15 displays the different streams that were being de-multiplexed after it was decoded by the system. There were a total of 6 streams composed of 3 video streams and 3 audio streams. The video streams that were decoded had a format of H264 or MPEG-4 AVC, while the audio streams had a format of MPEG AAC Audio (MP4A) which validates the specified standards of the ISDB-T transmission.



```
Current media / stream statistics
▾ Audio
    Decoded                     2614 blocks
    Played                      2614 buffers
    Lost                           0 buffers
▾ Video
    Decoded                     2754 blocks
    Displayed                   2641 frames
    Lost                           9 frames
▾ Input/Read
    Media data size           247612 KiB
  ▸ Input bitrate              17852 kb/s
    Demuxed data size         230105 KiB
    Content bitrate            16171 kb/s
    Discarded (corrupted)         10
    Dropped (discontinued)         0
▾ Output/Written/Sent
    Sent                           0 packets
    Sent                           0 KiB
    Upstream rate                  0 kb/s
```

Fig. 16. Current Media / Stream Statistics of TS

The figure above displayed the statistical values of the decoded signal. The input bit rate of the transport stream file was 17.852 MB/s. For the Audio steam: 2614 Audio blocks were decoded, 2614 audio buffers were played and a value of 0 lost buffers. For the Video stream 2754 Video blocks were decoded, 2641 video frames were displayed and 9 frames were lost. The .ts file displayed a high-quality video and a clear audio. Given such data, the raw baseband samples were properly decoded by the system.

B. *Reception of Free-to-air ISDB-T signal*

Shown in Figure 17 is the frequency spectrum display of the 6MHz bandwidth of the received ISDB-T signal of ABS-CBN Channel 37 UHF. Centered at 611.143MHz, the bandwidth attains a peak power of an estimated -40dBm. The obtained peak power value of around -40dBm validates that the system is successful in the reception of a real time ABS-CBN ISDB-T signal since it is determined that a power value of at least -62.66dBm indicates that the primary user (ABS-CBN) is transmitting ISDB-T signals [10]. As of present, ABS-CBN is the only station with free-to-air digital ISDB-T signal reception in Cebu.

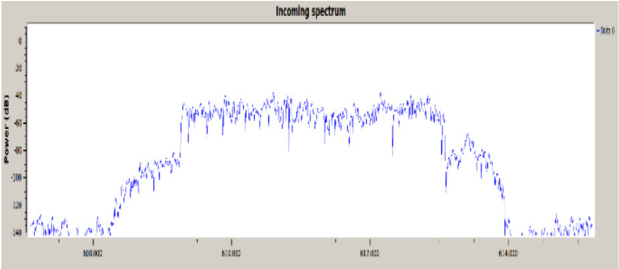


Fig. 17. Signal Spectrum of Received Signal

V. Conclusion

The ISDB-T decoder system was successfully implemented through simulation in GRC and through partial reception of a locally transmitted ISDB-T signal in software defined radio platform using URSP1 interfaced with the GNU Radio Companion installed in a host Linux PC. A raw baseband ISDB-T signal was tested and simulated through the configuration of the different signal processing blocks in GRC. With the file sink block as the final stage of the GRC flowgraph, a transport stream .ts file stored all decoded data and by the use of VLC media player, a simultaneous video display of the decoded signal was displayed. The simulation output also presents the frequency spectrum of the ISDB-T signal having a bandwidth of 6MHz which had a center frequency of 569MHz and a clearly presented constellation diagram with 64QAM as the modulation scheme. The guard interval configuration of the decoder was set at 1/16. A measured Modulation Error Rate (MER) of 34.22dB was obtained and this implies that the reception of the modulated signal is excellent. The measured Bit Error Rate (BER) of 8.95 for Viterbi decoder and 18.0 for Reed Solomon decoder implies that very minimal error was obtained by using the convolutional rate of 3/4. Setting the guard interval to that of 1/4, 1/8, or 1/32 and a convolutional coding rate of 1/2, 2/3, 5/6, or 7/8 resulted to a noisy reception, thus a negative result. To completely decode an ISDB-T signal, it is vital to determine the transmission parameters being configured beforehand.

For the partial reception of the local ISDB-T signal transmission in software defined radio platform, the system was able to display the 6MHz bandwidth of the ABS-CBN ISDB-T signal centered at 611.143MHz. A peak power value of around -40dBm was obtained and this implies that the system is able to receive a real-time ISDB-T signal. However, the decoding process was only partially implemented due to the proprietary Digital Television Scheme of the ABS-CBN channel restricting full disclosure of its transmission parameters thereby resulting to incomplete or partial decoding.

The host PC plays a critical role in the process of conducting this study and the researchers have realized that it is important that the computing machine should have a core processor of at least intel core i5 clock @ 2.7GHz for it would greatly affect the decoding process. A deficiency in processor performance caused underruns in the execution of the GRC flowgraph.

Acknowledgment

The authors would like to thank the Office of Research and the Department of Electrical and Electronics Engineering of the University of San Carlos for providing the funding support, and the necessary equipment and space during the conduct of this study. The following people also provided moral and technical support that helped a lot in the accomplishment of this research namely Gerome Jan Llames, Theodore James Romo, and Federico La Rocca.

References

[1] National Telecommunications Commission, “Standard for Digital Terrestrial Television (DTT) Broadcast Service,” [Online]. Available: http://www.ntc.gov.ph/laws/mc/1383722723_MC%2005-11-2013.pdf

[2] C. Tolibas, J. Naval, M. Donn, Generation of a Modified Integrated Services Digital Broadcasting Terrestrial (ISDB-T) Signal Using Software-Defined Radio, Cebu, 2014.

[3] F. S. Caluyo and J. C. Dela Cruz, “Antenna Characterization and Determination of Path Loss Exponents for 677MHz Channel using Fixed and Portable Digital Terrestrial Television.” *Progress in Electromagnetics Research*, Volume 29, 2012, Pages 149–161.

[4] F. Larroca, P. F. Guridi, G. G. Sena, V. G. Barbone, and P. Belzarena, “gr-isdbt: An ISDB-T 1-segment Receiver Implementation on GNU Radio.” 2015 Latin American Computing Conference (CLEI).

[5] ARIB, “Transmission System for Digital Terrestrial Television Broadcast,” [Online]. Available: http://www.arib.or.jp/english/html/overview/ov/std_b31.html

[6] Y. Takahashi, “Transmission System,” June 2007. [Online]. Available: http://www.dibeg.org/news/previous_doc/0706_2Brazil_ISDB-T_seminar/Brazil-ISDBTseminar3-transmission_system.pdf

[7] M. S. M. Takada, “Transmission System for ISDB-T,” [Online]. Available: http://www2.elo.utfsm.cl/~elo352/Experiencia%20TVD%20web/PaperTVD.pdf

[8] DiBEG, “Features of ISDB-T system,” 2007. [Online].

[9] J.-P. Lang, “GNU Radio,” 2006–2013. [Online]. Available: http://gnuradio.org/redmine/projects/gnuradio/wiki.

[10] A. B. G. J. Llames, Spectrum Sensing System in Software-Defined Radio for Determining Spectrum Availability, Hanoi, Vietnam, 2016.

[11] T. V. Wilson, “How HDTV Works,” [Online]. Available: http://electronics.howstuffworks.com/hdtv2.htm

[12] E. Tse, “Software Radio for Digital Satellite,” 17 November 2010. [Online]. Available: http://www.edmundtse.com/wp-content/uploads/2009/04/treatise.pdf

[13] Got Questions Ministries, “What is the human soul?,” 2000-2016. [Online]. Available: http://www.gotquestions.org/human-soul.html

[14] I. Poole, “What is DVB-T?” [Online]. Available: http://www.radioelectronics.com/info/broadcast/digital-video-broadcasting/what-is-dvb-t-basics-tutorial.php.

[15] Ali Hamzi, in *Studies in Digital TV Signal Processing: Impulse Noise Mitigation, Repeater Loop Interference Cancellation, and DVB-T Transmission in CATV Networks*, Jukaisu 668, Publication 668.

[16] H. Sugano, R. Miyamoto, M. Okada, "Fully Software-based real-time ISDB-T 1 segment receiver," [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5954956

[17] J. Song, Z. Yang, L. Yang, K. Gong, C. Pan, J. Wang, and Y. Wu, "Technical Review on Chinese Digital Terrestrial Television Broadcasting Standard and Measureements on Some Working Mode," *IEEE Transactions on Broadcasting*, *Volume* 53, *Issue* 1, March 2007.

[18] "DTV Status," [Online]. Available: http://www.dtvstatus.net/map/map.html

[19] DVB Project Office, "Digital Terrestrial Television," August 2012. [Online]. Available: https://www.dvb.org/resources/public/factsheets/DVB-T_Factsheet.pdf

[20] Commission on Higher Education, "CHED Memorandum Order (CMO) No. 24 Series of 2008," 2008.

[21] Ming Liu, Matthieu Crussière member IEEE, Jean-François Hélard senior IEEE, Oudomsack Pierre Pasquero, "Analysis and Performance Comparison of DVB-T and DTMB Systems for Terrestrial Digital TV," 11th IEEE Singapore International Conference on Communication Systems, 2008 (ICCS 2008).

Efficient Load Balancing Technique for Parallel Ray Tracing Using a Reservoir

Joy Alinda Madamba and Francis Joseph Seriña

***Abstract*—Imitating the real world in emulations, such as the effect of light, is an important research area in software applications. However, a computing machine cannot precisely imitate all the aspects of light in short processing periods. Ray tracing is a technique that renders and accurately depicts light with a high time delay. Parallelizing the computers that run the rendering is not enough to reduce its running time. To further increase computing efficiency and improve ray tracing's applicability for mainstream purposes, load balancing must be performed. In this study, a proposed alternative load balancing technique that uses a reservoir was implemented to tackle the issue of using heterogeneous computers in the network and to minimize the communication overhead introduced by parallel applications. Results show that when using a reservoir, high speed-up is achieved at 80% initial task distribution while high efficiency is achieved between 20–75% initial task distribution. With speed-up as the priority, reservoir achieves at most 87% better speed-up than the scatter decomposition and 36% than the hybrid technique. It also has at most 28% better efficiency than the scatter decomposition. Finally, the reservoir algorithm achieved at most 25% better achieved speed-up (ASU) than the scatter decomposition and 10% than the hybrid.**

***Index Terms*—Computer Graphics, Load Balancing, Parallel Programming, Ray Tracing, Rendering**

I. Introduction

THE recent trend in computer development has been towards shorter execution times for complex applications. However, these applications have developed faster than the advances done in computer architectures. Single-core processors have thus given way to multi-processors, paving the way for parallel solutions to these complex problems. An example of this complex application is ray tracing. Many people consider ray tracing to be the best image synthesizing technique to date [1]. Ray tracing naturally shows precise reflections, shadows, and transmissions by following the path of light and applying the laws of geometric optics. Unfortunately, high quality images can only be produced at a large amount of time. Ray tracing has not been used in mainstream graphics yet, aside from the film industry, because no computing machine can single-handedly render large images in a relatively small useful amount of time [2]. Thus, ray tracing is a good candidate for implementation in a network of computers that would execute that application in parallel.

The use of parallelization has its own set of areas that need improvement when it comes to efficiency. One such issue is load balancing. The workloads assigned to different computers are not always equal and a balancing scheme must be applied in such a way that the computers complete their task at the same time [3]. Load balancing techniques are classified as static and dynamic. Static load balancing distributes tasks before processing, taking into account certain assumptions such as tasks with equal workload unless otherwise stated. Static load balancing techniques are based on the partitioning of the image plane (screen). One static technique found to be effective for network systems with less than 128 computers is Scatter Decomposition [4], [5]. In this algorithm, the tasks are alternately distributed to increase the probability that machines get equal workload amounts. It distributes the pixels in an alternating sequence ($i \bmod p$) where the p is the number of computers and i is the pixel to be rendered. The result of the alternating sequence formula is the computer, which renders that pixel. For example, Figure 1 shows how scatter decomposition is applied unto a picture with 100 pixels per row. Given that there are 3 computers within the network, the 1st, 4th, 103rd, 106th, etc. pixel is assigned to the first computer while the 2nd, 5th, 101st, 104th, etc pixel is given to the second computer and finally, the 3rd, 6th, 102nd, 105th, etc. pixel is given to the third computer.

J. A. Madamba is with the Electrical and Electronics Engineering Insitute, University of the Philippines, Diliman (e-mail: joyarmadamba@gmail.com)

F. J. Seriña is with Autodesk Inc., at Pittsburgh, Pennsylvania

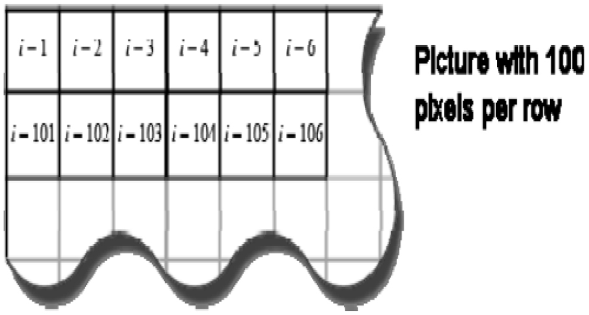


Fig 1. Scatter decomposition

Dynamic load balancing redistributes the tasks during processing. Techniques that use this type of load balancing can be further subdivided into two approaches: centralized and decentralized [6]. The centralized approach uses one machine as the controller (or master) of the system. It is responsible for task allocation and assignment. The decentralized approach uses all the machines in deciding where the next tasks will be given, with or without cooperation between machines. We focused on the centralized techniques as these would be more suitable for ray tracing. Some centralized techniques include the algorithm by Wang, et al. [7] that assigns tasks based on whether it is an I/O consuming task or a CPU compute task and the algorithm by Sidhu et al. [8] that uses particle swarm optimization. However, the first algorithm mentioned was not used as the assignment of tasks would introduce more variables in a heterogeneous system and the second is more suitable to a large-scale network. A third highly scalable centralized dynamic technique is Diffusion, which redistributes the tasks evenly when one of the slaves have completed earlier than the others [4]. Diffusion is started when a slave completes its task queue by sending a request signal to its neighboring machines. The neighbors reply with the number of tasks left in their queue. The average number of the remaining tasks is determined and the machines with a higher average give their excess tasks to the machines with a smaller queue. It has been shown that a hybrid technique (scatter decomposition with diffusion) resulted in greater efficiencies and speed up compared to them working alone [4]. However, scatter decomposition and diffusion only used homogenous computer architectures in their networks.

In order to normalize the performance of networks that use computers with heterogeneous architectures, an additional parameter is taken into consideration: speed index [5]. This determines the relative speed of one machine over another within the system. It is measured by determining the execution time of running a recursive function. Thus, distribution of tasks took advantage of which computer is faster by possibly assigning more tasks to these computers.

The goal of the study was to implement a new load balancing technique that uses a reservoir and compare its performance against existing load balancing algorithms. The study was implemented on a fully connected network with different computer hardware architectures. Figure 2 shows the hardware environment and the communications involved in the system. The hardware environment used heterogeneous commercial desktop computers with 1 designated as the master and the rest as slaves. It used the distributed memory architecture and communicated over a fully connected network.

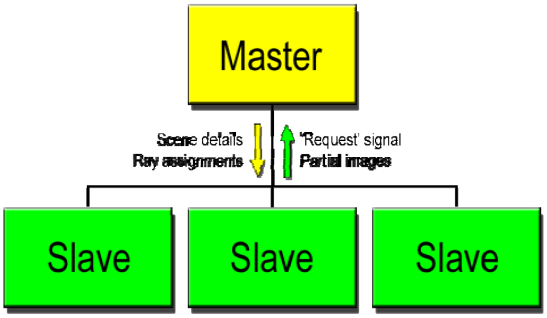


Fig 2. Implementation environment

The master was responsible for initializing the scene and distributing the tasks. It shall accept a 'request for more' signal from any slave and give it more tasks to do unless there's none left. The master accepts all the partial images from the slaves and compiles them as a single image. The slaves performed the ray tracing and requested for more tasks once its queue is empty.

II. Methodology

A. Reservoir Algorithm

The diffusion algorithm has a high communication overhead especially when the computers are in a fully connected network. When a slave requests for more rays, the master will ask each slave to send their task queue. The master averages the number of tasks and calls the slaves with task queue higher than the average to distribute their tasks to those who have less. Meaning, it redistributes tasks that were initially assigned already. This overhead is neglected since the amount of time it takes to send and receive data is much smaller than the rendering process. The reservoir algorithm will reduce the communication overhead by preventing any redistribution needed in the middle of the process. It does this by making the initial task distribution biased in such a way that faster computers receive more. Also, it does not distribute all the tasks during initial assignment. It keeps a certain number of tasks in a reservoir which will only be assigned when a slave requests for more.

There are three (3) major blocks in the Reservoir algorithm, as shown in Figure 3. The Reservoir Preparation uses the speed index of each slave to calculate the division of labor when biasing the task distribution. The steps for this procedure are shown next.

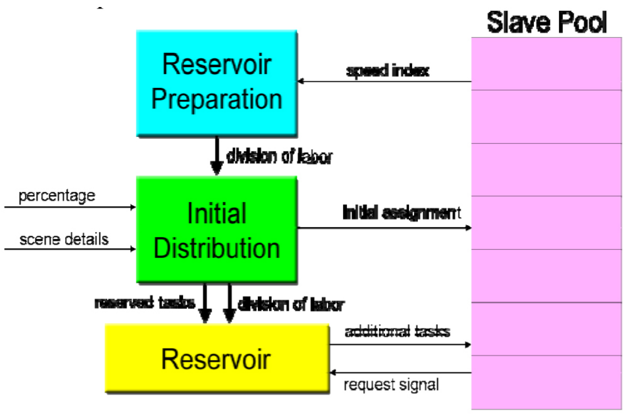


Fig. 3. Reservoir block diagram

Input: speed indices ($speed_1$, $speed_2$, ..., $speed_c$) and the number (c) of slaves
Output: sequence of percentage division of labor per slave (div_1, div_2, ..., div_c)

```
1.  procedure reservoir_preparation (speed, c)
2.          slowest : = $speed_1$
3.          for i : = 2 to c do
4.                  if $speed_i$ > slowest then
5.                          slowest : = $speed_i$
6.          ratio_total : = 0
7.          for i : = 1 to c do
8.                  begin
9.                  $ratio_i$ : = slowest / $speed_i$
10.                 ratio_total : = ratio_total + $ratio_i$
11.                 end
12.         for i : = 1 to c do
13.                 $div_i$ : = $ratio_i$ / ratio_total
14.         return (div)
15. end reservoir_preparation
```

The variable slowest is the speed index of the slowest computer within the network. This is searched for linearly in lines 2–5. In the sequence, ratio$_i$ is the theoretical equivalent amount of work that can be completed by slave$_i$ in the same time that the slowest slave completes 1 task—assuming that all tasks are equal. The sum of all ratio's are stored in ratio_total (lines 6–11) to be used as divisor in lines 12–13 to determine the division of labor.

The Initial Distribution block initially allocates specific amounts of the total tasks to the slaves. This initial number of tasks is determined by a quantity specified by the user called the initial task distribution percentage or initial percentage. Subsequently, a sequence of pixels is computed by multiplying the division of labor, initial percentage, and total number of tasks. These pixels are then given in a sequential manner. The first sequence of pixels is given to the first slave, the second sequence to the second slave, etc. The remaining unassigned tasks will be placed in the last block as reserved tasks. The steps for this procedure are shown next.

Input: initial task distribution percentage (p)
sequence (ray_1, ray_2, ..., ray_n) and number (n) of tasks
sequence ($slave_1$, $slave_2$, ..., $slave_c$) and number (c) of slaves
division of labor (div_1, div_2, ..., div_c) per slave
Output: undistributed tasks ($reserve$)

```
procedure initial_distribution (p, ray, n, slave, c, div)
limit : = p * n
prev_limit : = 0
for s : = 1 to c do
        begin
        next_limit : = prev_limit + (limit * $div_s$)
        for i : = prev_limit to next_limit do
                begin
AssignTask ($ray_i$, $slave_s$)
                end
        prev_limit : = next_limit + 1;
        end
reserve : = $ray_{limit+1}$, $ray_{limit+2}$, ... $ray_n$
        return (reserve)
end initial_distribution
```

When a slave finishes its assigned tasks, it will ask for more from the reservoir. In the final block of the algorithm, a fraction of the reserved tasks is given to the requesting slave based on the division of labor. The algorithm ends when there are no more reserved tasks, as shown below.

Input: requesting slave ($slave$)
division of labor of requesting slave (div)
sequence (ray_1, ray_2, ..., ray_n) and number (n) of undistributed tasks
Output: remaining undistributed tasks ($reserve$)

```
procedure Reservoir (slave, div, ray, n)
        limit : = div * n
        for i : = 1 to limit do
                AssignTask (rayi, slave)
        reserve : = raylimit+1, raylimit+2, ..., $ray_n$
        return (reserve)
end Reservoir
```

TABLE I
Sample Simulation

Slave	Speed Index	Slowest	Ratio	Sum of Ratios	Percent division of labor	Ex 100 Tasks
1	1	Slave 3 with speed index 12	12	18.714	64.12% (65%)	65
2	3		4		21.37% (22%)	
3	12		1		9.16% (10%)	
4	7		1.714		5.34% (6%)	

All arithmetic operations are rounded up for the algorithm to converge. A test simulation is shown in Table 1. Slave 3 is determined to be the slowest with speed index 12. Ratio is the slowest speed index divided by the speed index of the slave under consideration. It represents how many tasks that computer can accomplish compared to the slowest assuming that all tasks are equal. For example, since slave 1 is 12 times faster compared to the slave 3, it could theoretically complete 12 tasks when slave 3 completes 1. The sum of these ratios will be used as a divisor to acquire the percent division of labor. The division of labor must be rounded up for the algorithm to converge.

On the last column, we assume slave 1 finished its tasks and there are 100 tasks in the reserve. Since the percent division of labor of slave 1 is 65%, 65 tasks will be given to it and the rest will remain as reserve until another slave completes its queue.

B. *Phase I – Serial Ray Tracer (SRT)*

In the first phase of the study, a functional serial ray tracer was built, which was used as the basis for the succeeding tracers. Ray tracing is a pixel by pixel type of rendering. Each pixel in the final image is calculated independently from those beside it. Looking at Figure 4, **E** refers to the viewer's eye which is looking at object **6**. Objects **6** and **4** are opaque spheres. Objects **3** and **9** are translucent planes and **LA** and **LB** are light sources. The different rays will be described later. A primary ray (from **E** to **3**) from the viewport travels in a straight line towards a pixel on the screen. This ray continues to the scene until it hits a surface. The algorithm that finds any intersection is called visible surface determination or intersection tests. Different algorithms were developed to find the most efficient visible surface determination algorithm. This is often a bottleneck in ray tracing algorithms as well as trying to parallelize it. Characteristics of the surface are taken into consideration to determine the reflection and refraction properties as well as its color at the point of intersection.

Shown in Figure 5 is the serial ray trace flowchart. The shading model used in the flowchart is a simplified hall shading model which consists of 4 terms. The first term is the low-level ambient light. The second term is a modified Phong Shading model which combines diffuse and specular term. The ambient term is neglected because the simplified hall shading model already considers it. The third term is the perfect specular reflection and the fourth is perfect specular transmission. The third and fourth terms are subject to recursion.

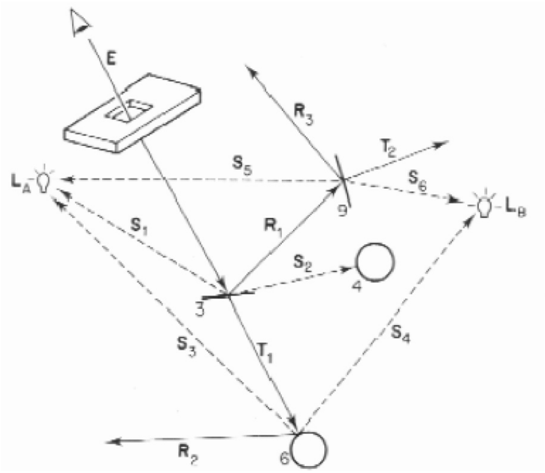


Fig. 4. Ray tracing paths

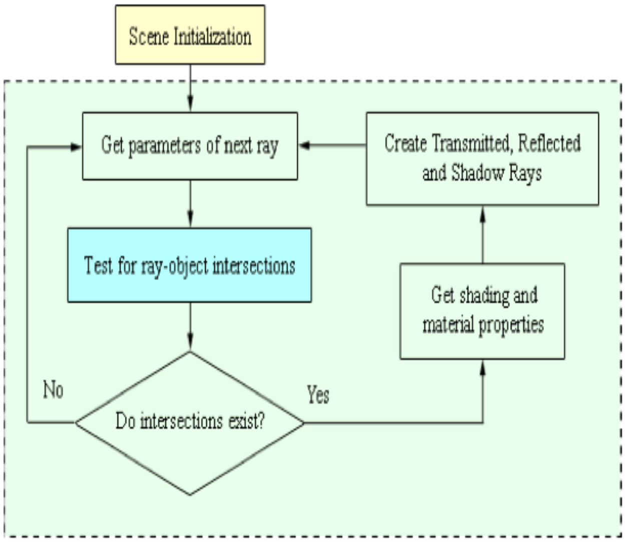

Scene Initialization
Get parameters of next ray
Create Transmitted, Reflected and Shadow Rays
Test for ray-object intersections
Get shading and material properties
No
Yes
Do intersections exist?

Fig. 5. Serial ray tracer

To make ray tracing faster, effort was given to the intersection test algorithm. The bulk of the running time of any ray tracer is devoted in finding the intersection between each ray and scene objects. To reduce the calculations, each object is assigned a bounding box—it is a box that tightly covers the space occupied by each object. The minimum and maximum x, y, and z coordinates of the object will be used as the dimensions of the bounding box. The 6 coordinates will correspond to the 6 faces of the box. When testing for the intersection between a ray and an object, the ray is tested against one face at a time.

The ray tracer could read the scene details from an external text file. The details include camera information, output image information, light sources, object (with surface) properties and environment variables. The output image information includes the width and height in pixels and the output file name, which has the extension of the output image type. The light sources are defined using its location and color. The object properties are defined differently. Surface properties have the same syntax for all types of primitives and are defined together with the object. The environment variables include the background color, ambient color and the maximum levels of reflection/ transmission rays.

The serial ray tracer, using the minimum set of features, was able to render images with a single machine. It was able to render three (3) different types of primitive objects, namely spheres, boxes and one-sided planes, shown in Figure 6. Its resulting images have shown reflections, transmissions and shadows, similar to Figure 7.

Additional features such as super sampling and distributed ray tracing were also implemented to improve the quality of the pictures and increase the weight of the tasks. Super sampling is an anti-aliasing technique by shooting more rays per pixel. Distributed ray tracing slightly changes the angle of produced reflection, refraction and shadow rays to apply blurry phenomena such as gloss, blurry transparency, and penumbras. Shown in Figure 8 is a sample scene that shows all the basic features. A transparent sphere with properties similar to glass is included in the figure. Below it is a plane that reflects everything on top of it. There are multiple opaque left objects, 3 smaller blue spheres and a yellow box behind the transparent sphere. Two point-light sources are used, one behind the camera and another to the left of the box. Super sampling and distributed ray tracing were also implemented. The functionality of the SRT was then tested and verified.

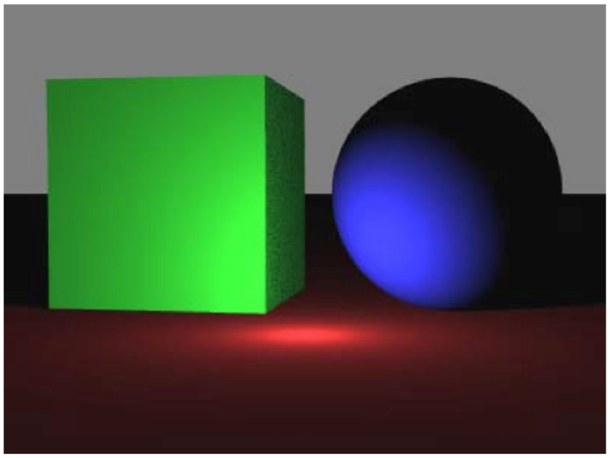


Fig. 6. Basic primitives

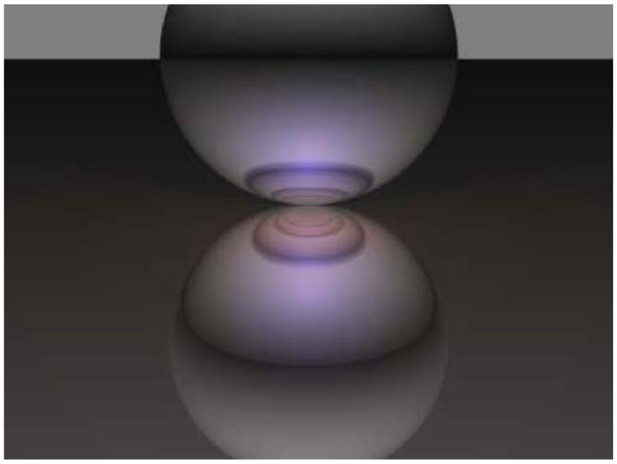


Fig. 7. Shiny sphere over a mirror

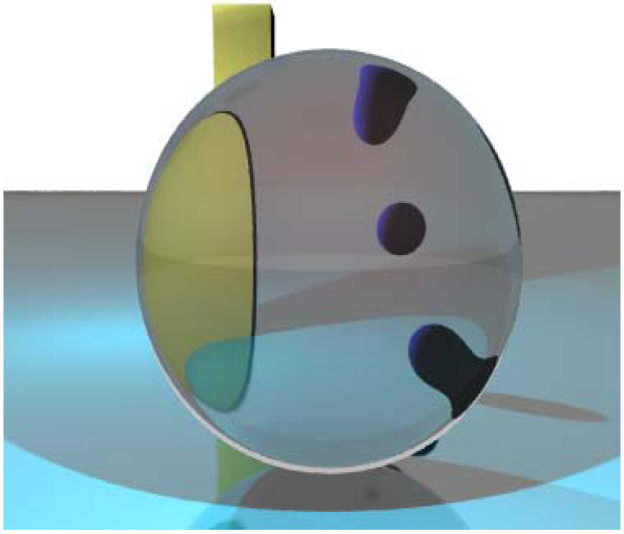


Fig. 8. Image created by SRT with basic features

C. *Phase II – Parallel Ray Tracer*

The next phase involved implementing the first of the three (3) parallel ray tracers (PRT). This PRT implemented only the scatter decomposition as a load balancing technique.

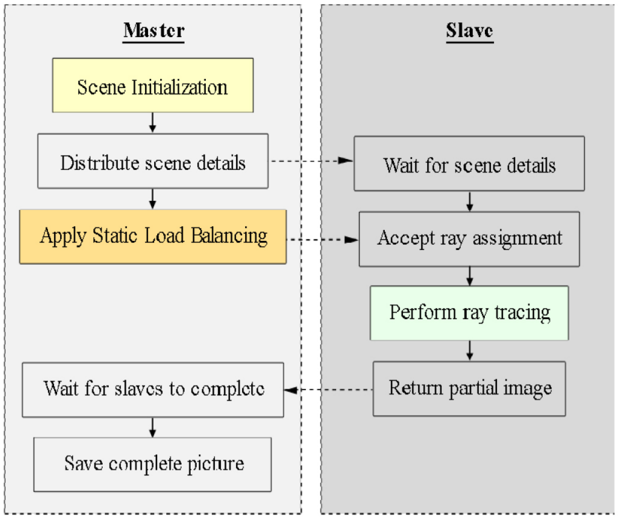


Fig. 9. Parallel ray tracer

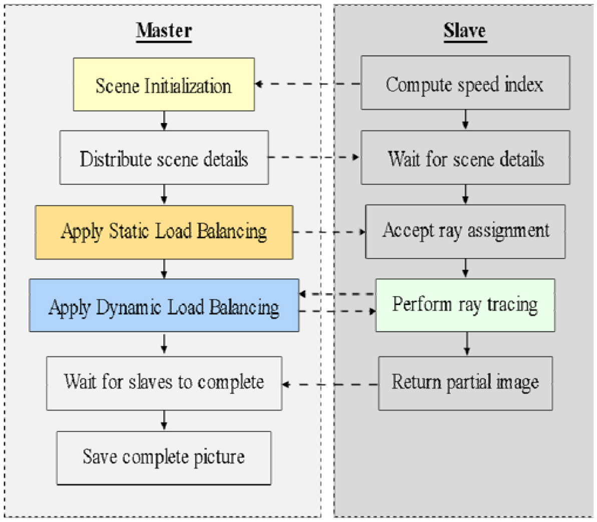


Fig. 10. Parallel ray tracer that uses dynamic load balancing

A master and slave side was created to distribute the tasks of the PRT. Additional modules were created for the master to pass the scene details to the slaves and for the slaves to compile the partial images for the master. The flowchart and points of communication are shown in Figure 9. Initialization is executed by the master while the slave executes ray tracing. The distribution of rays amongst the slaves, third block in the master side, is the static load balancing.

Different computers were used to test the parallelized ray tracer. The execution and idle time of each slave are noted in order to calculate efficiency and speed up. The results are tested against the results of phase III. The overall efficiency of the program is the sum of the execution times of each slave over the total elapsed time (execution and idle time) of all the slaves.

D. Phase III – Parallel Ray Tracer with Dynamic Load Balancing

The last two PRTs were implemented in this phase. The first PRT used the hybrid load balancing technique which combines scatter decomposition and diffusion. The second PRT implemented used the reservoir algorithm. Additional modules were also created, such as the module that computes the speed index and the module that commands slaves to receive/send tasks from/to other slaves. Figure 10 shows the block diagram of the PRT with dynamic load balancing.

E. Testing and Benchmarking

For the testing of the SRT, a series of test scenes were created that focused on the specific features of the ray tracer. This set of scenes was also rendered using an older version of a commercial ray tracer originally developed by Mental Images, the Mental Ray tracer. The resulting images were compared against those created by the SRT.

The test scenes created focused on the following features: shading, reflections, shadows, transmissions, and their combinations. Figure 11 shows one image that combines all of the features.

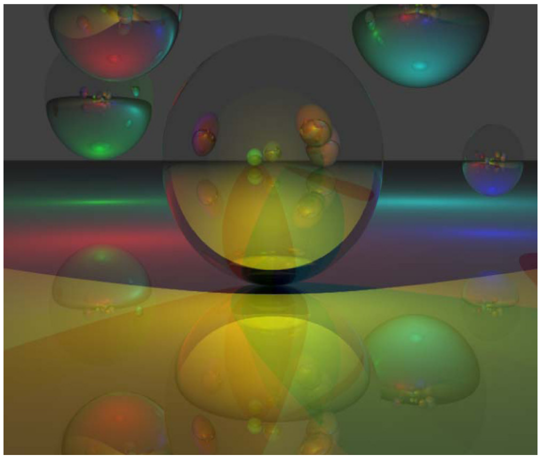


Fig. 11. Test scene that combined features

To compare the three (3) PRTs with the SRT, a complex scene (Figure 12) was rendered with varying parameters by all four ray tracers. This test scene approximated practical scenes since it has varying workloads throughout the image. The SRT was modified to calculate its speed index even if it was not used during its image processing. It was used later as a reference for comparison. For the PRTs, the total program execution times and slave speed index were recorded. The speed-up was computed by dividing the execution time of the PRT over the execution time of the SRT that rendered the image with the same parameters. It was assumed that since the hardware environment used was exactly the same for all the tests, the speed indices of the machines within the network would be the same for all tracers.

The efficiency is computed as follows:

$$\eta = (\sum ET_S)/(ET_T \times n) \quad (1)$$

where ET_T is total elapsed time of the program, n is the number of slaves, and ΣET_S is a summation of the execution times of all slaves.

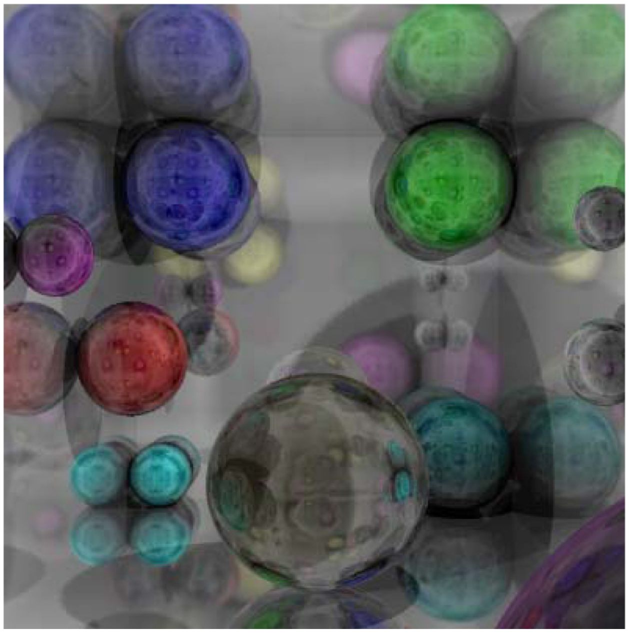


Fig. 12. Complex scene used as a benchmark.

Since the computers within the network do not have equal speeds, a new unit of measurement was used. This is the achieved speed-up (ASU) which relies on the validity of the speed index. The ASU is the ratio of the computed speed up compared to the maximum achievable speed up (MASU). Table 2 shows an example on how to compute the MASU and ASU. The theoretical speed-up is computed by dividing the speed index of the SRT over the speed index of the machine under consideration. The sum of all these theoretical speed ups would account for the MASU. If all the machines are homogenous, including the SRT, then the MASU is equal to the number of the slaves (ex. 4 homogenous slaves have 400% MASU). For the example below, the MASU is 341.79%. The ASU is then computed by dividing the speed up over the MASU.

TABLE 2
Sample Computation of ASU

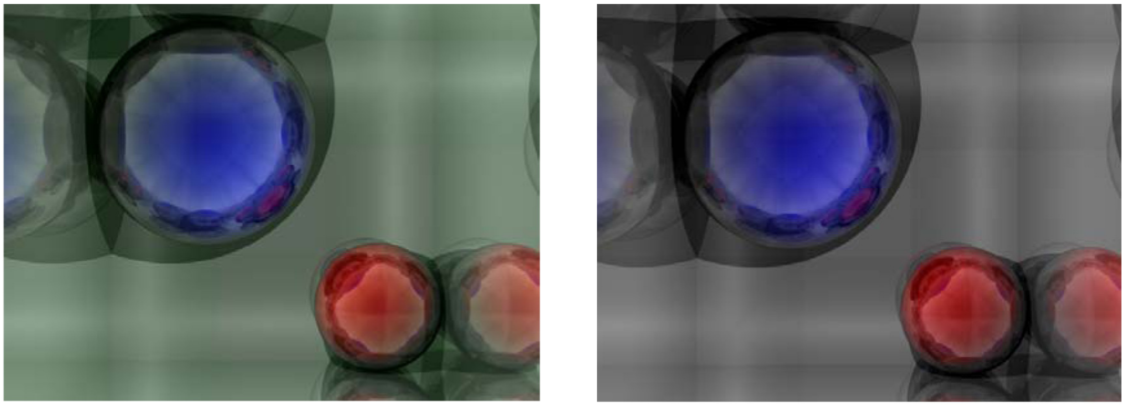
Machine	Speed Index (seconds)	Theoretical Speed Up (%)	Achievable Speed Up (%)
SRT	0.3	100	–
Slave 1	0.30525	98.28	28.75
Slave 2	0.644833	46.52	13.61
Slave 3	0.301333	99.56	29.13
Slave 4	0.307917	97.43	28.51

III. Results and Analysis

A. Validity of Features

The SRT built from phase I was compared with another ray tracer using special scenes that were customized in order to exclude other features that the serial ray tracers could not perform. The following are a series of images of the different scenes rendered in both the SRT and Mental Ray Renderer.

As can be seen, Figure 13 has a significant difference—mainly, the hue of green visible in the whole scene. This is



a. Mental Ray Render

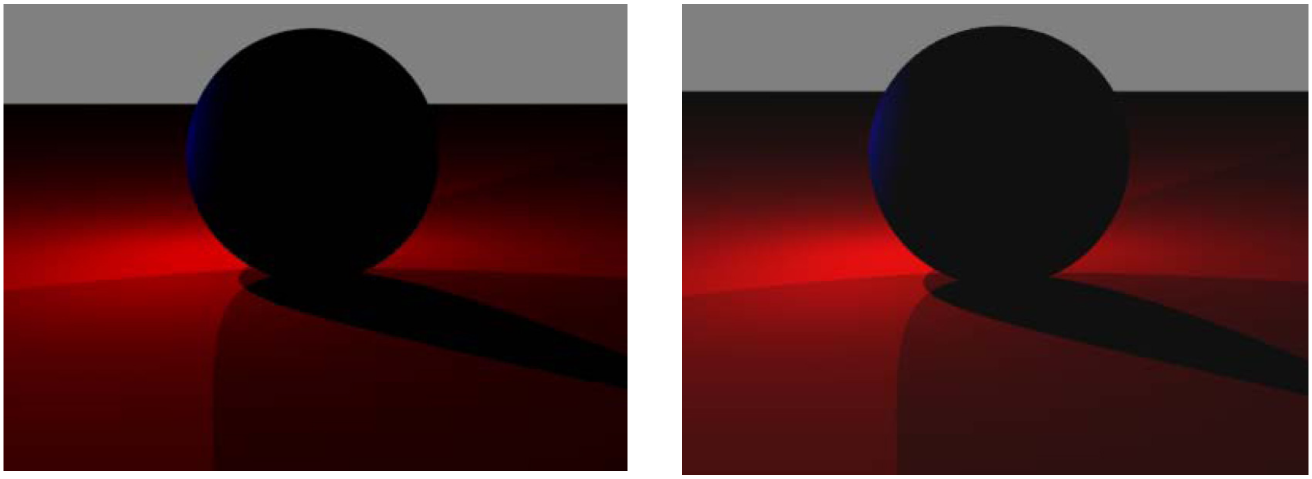
b. Serial Ray Tracer

Fig. 13. Shading equation and reflections

due to the set ambient light and the shading equation used. The Mental Ray Renderer applies a bigger weight to the ambient color compared to the serial ray tracer. Looking at the reflection and the color of the objects in the scene, it can be shown that other elements of the shading equation between the Mental Ray Renderer and the serial ray tracer is the same for the surface properties present.

In Figures 14 and 15, we can clearly see that the images produced are identical, showing the accuracy of the serial ray tracer in depicting shadows and transmissions.

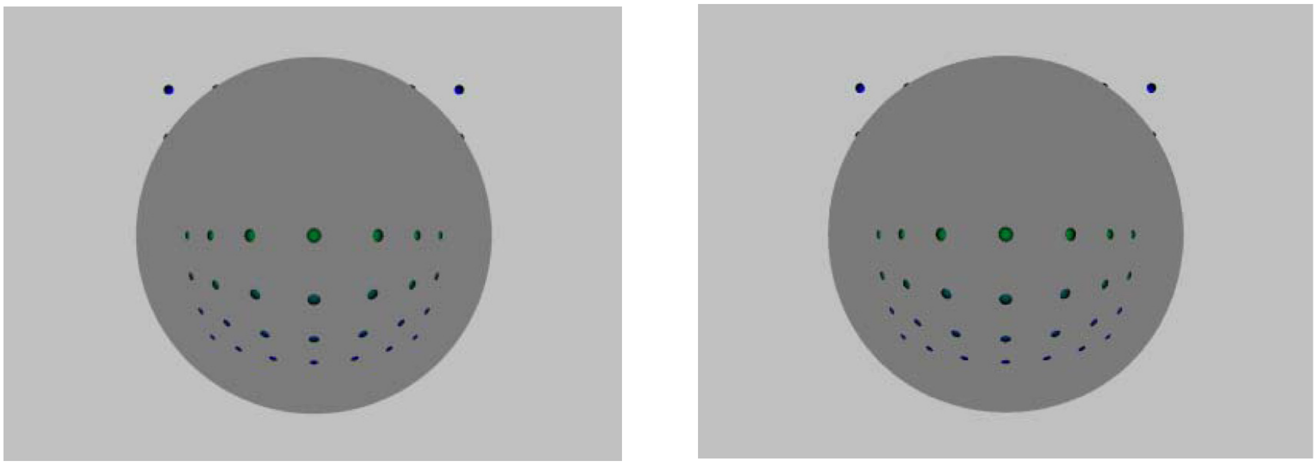
Figure 16 shows that when the scene gets more complicated, the difference between the serial ray tracer and that of the Mental Ray Renderer becomes unnoticeable.



a. Mental Ray Render

b. Serial Ray Tracer

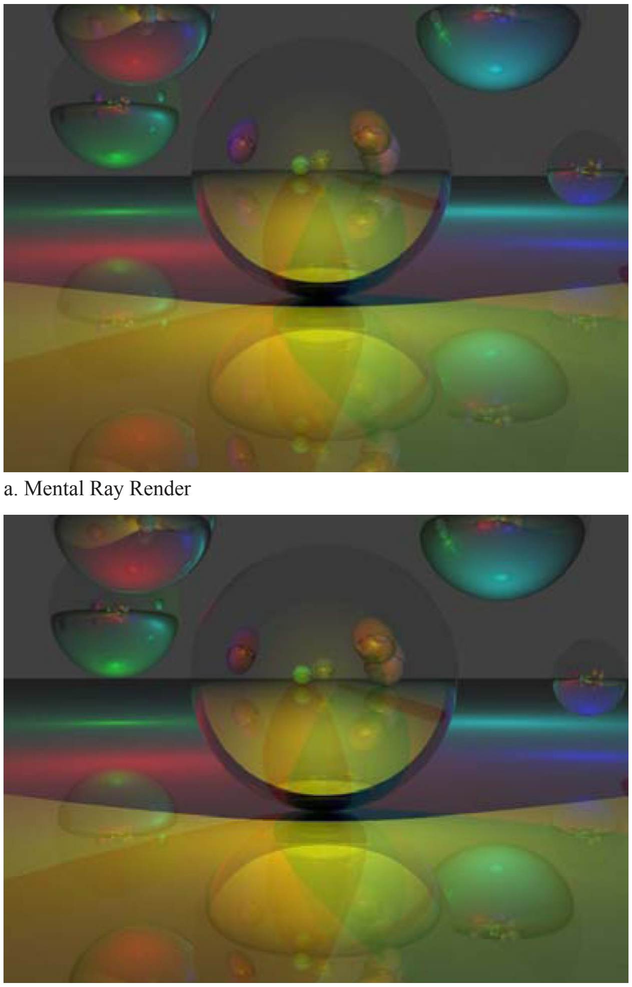
Fig. 14. Shadows



a. Mental Ray Render

b. Serial Ray Tracer

Fig. 15. Transmissions



a. Mental Ray Render

b. Serial Ray Tracer

Fig. 16. Lights, reflections and shadows

B. *Serial Ray Tracer Execution Times*

One objective of this study was to build a PRT and compare the performance of the proposed algorithm with the existing load balancing techniques. Using the complex scene at Figure 12, Table 3 shows the execution time of the serial ray tracer, with an average speed index of 0.3, with different parameters. DRT refers to distributed ray tracing rate, SS is the super sampling rate, ET is the execution time in seconds. Since these four test cases were used all throughout the testing, they will be referenced as shown. The execution time is shown in HH:MM:SS. The four test cases do not have a constant increase in complexity of rendering with these parameters but as the execution time shows, test case C and D are much more difficult compared to A and B. This table will also serve as a reference for the speed up later on.

TABLE III

Serial Ray Tracer Results

Test Case	DRT	SS	ET
A	1	8	4:14.51
B	1	16	8:34.16
C	2	8	5:22:5
D	2	16	10:46:12

C. *Validity of Computing Speed Index*

To measure the speed indices of the machines used in the network, a recursive function, the classic Towers of Hanoi, was executed. This function was chosen due to its size. It is small enough to minimize the overhead for its computation and reliable enough to imitate the execution of ray tracing while measuring the speed index. To verify this assumption, the two algorithms were executed and compared using the speed up, MASU, and ASU. Reservoir was used as the load balancing technique with an initial percentage set to 80%.

It can be seen in Figure 17 that the difference of values in the MASU and ASU is at 23.81% and 3.14% respectively. This means that these speed index-based values for the two functions differ by less than 10% of the higher value (42.38 for the MASU and 5.31 for the ASU). Therefore, the method used to approximate the speed index is relatively accurate.

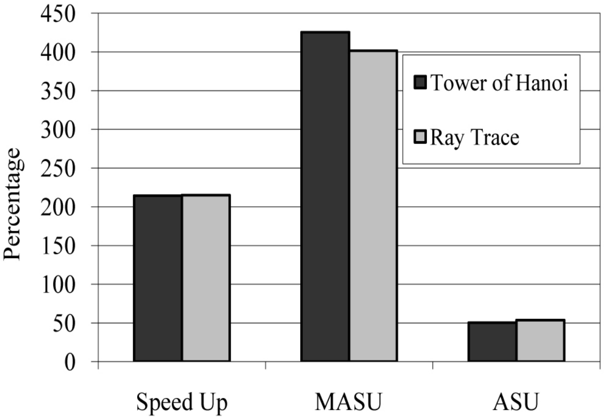


Fig. 17. Speed Indices of Algorithms

D. *Maximum Speed Up for Reservoir*

One difference of the reservoir algorithm over other load balancing techniques is the use of an additional parameter —initial task distribution percentage. In this test, the most effective initial percentage is determined for later use. The complex scene is rendered using test case C from Table 3 while varying the initial percentage. The speed up and efficiency of the PRT are shown in Figure 18.

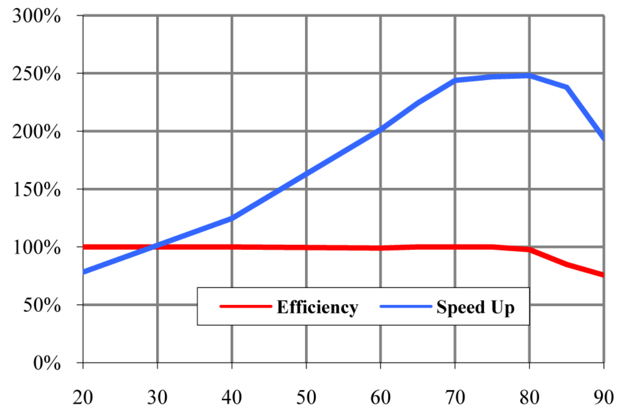


Fig. 18. Comparison of Efficiency and Speed Up of PRT Using Reservoir

Since the objective of parallelizing applications is to increase speed up, it can be shown that the best initial task distribution percentage for this environment is at 80%. Beyond 80%, the speed up decreases and it performs more similar to scatter decomposition both in efficiency and speed up. Below 80%, fewer tasks are initially distributed. These will make the slaves ask for more tasks earlier during the execution of the program. It is possible that the slaves tend to wait in line when requesting for additional tasks.

The efficiency, on the other hand, is consistently at 99% until the initial percentage is at 75%. Beyond that, it decreases. This is because the algorithm approaches the performance of the scatter decomposition implementation when using higher initial percentages. In these cases, a large number of tasks remain on the slaves and do not get redistributed even when the reservoir is empty and other slaves have completed their task queues.

The objective of parallelizing ray tracing is to shorten the execution time. It does not necessarily mean achieving the maximum efficiency. Thus, all PRT using reservoir used 80% as the initial task distribution percentage.

E. *Comparison of Parallel Ray Tracers*

Figure 19 shows the comparison of speed ups of the 3 PRT implementations. It can be seen in the results that the PRT with reservoir is better in most cases. The hybrid implementation is similarly high for the more difficult cases, approximately at 292% speed up. As expected, the PRT with scatter decomposition alone is the slowest of all implementations.

The speed up test is valid since the hardware environment is the same for all the different PRTs.

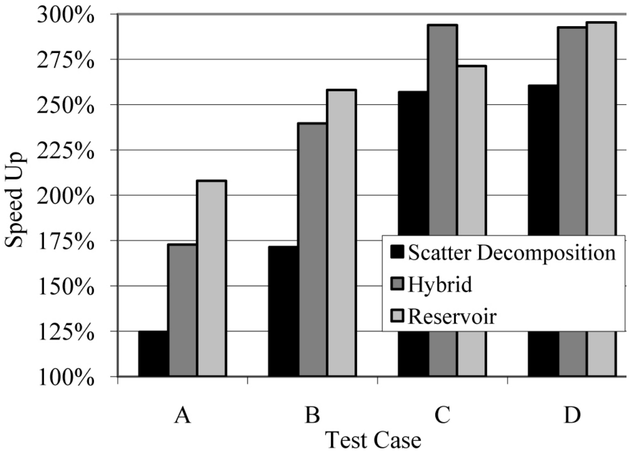


Fig. 19. Speed Up Comparison

Figure 20 shows the comparison of efficiencies. It can be seen in the figure that the hybrid implementation is consistently performing at around 99.9%, better compared to all three PRTs. However, the difference with the reservoir implementation is small, at less than 10%. This difference can be attributed to the initial task distribution percentage, which was not chosen for its maximum efficiency.

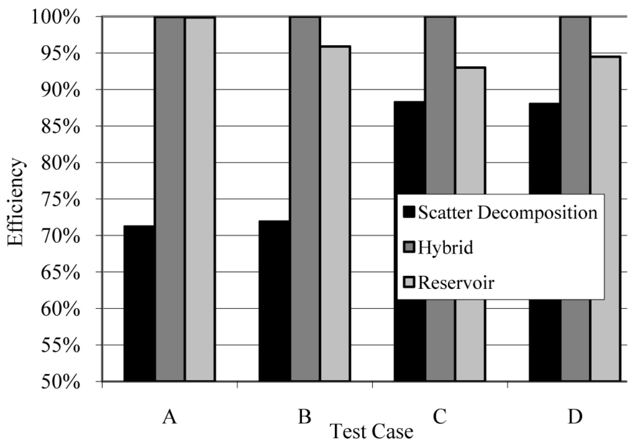


Fig. 20. Efficiency Comparison

Figure 21 shows the ASU of each PRT implementation. As seen in Figure 14, the reservoir implementation is better in most cases. This means that the algorithm was more able to maximize the hardware environment compared to the other implementations.

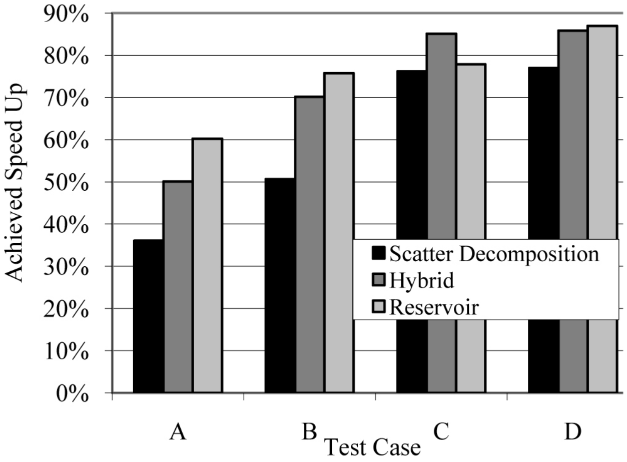


Fig. 21. Achieved Speed Up Comparison

IV. Conclusion and Recommendations

The results have shown that when using a reservoir, high speed-up is achieved at 80% initial task distribution while high efficiency is achieved between 20–75% initial task distribution. With speed-up as the priority, reservoir achieves at most 87% better speed-up than the scatter decomposition and 36% than the hybrid technique. It also has at most 28% better efficiency than the scatter decomposition. The reservoir algorithm achieved at most 25% better achieved speed-up (ASU) than the scatter decomposition and 10% than the hybrid. Thus, in conclusion, under the scenarios tested and the methodologies used, the load balancing technique that uses a reservoir is a good alternative technique for load balancing in parallel ray tracing. This technique has a simpler implementation than diffusion since assignment of tasks is only done once and no tasks are moved between slaves.

Even if complex scenes were used in testing, we recommend that real practical scenes be used as benchmarks. Personal evaluation of these scenes can be used as an additional test to determine the realism of the images. The concept of applying a bias to the distribution using the speed index can also be applied for other load balancing techniques that use heterogeneous computers. With this type of implementation, a possible increase to the speed up, ASU and the efficiency may be seen. Also, the current hardware environment is limited to a single master and four slaves. It is possible that the algorithm may perform differently with varying number of slaves. In this case, the ASU can be used as the primary measurement of comparison as the heterogeneity of the system will be more prominent.

References

[1] Glassner, A. *Introduction to Ray Tracing*. Morgan Kaufmann Publishers, Inc., 1989.

[2] J. Hurley. “Ray Tracing Goes Mainstream,” *Intel Technology Journal*. Volume 9, Issue 2, p. 107, May 19, 2005.

[3] Watts, J., Taylor, S. “A Practical Approach to Dynamic Load Balancing,” *IEEE Transactions on Parallel and Distributed Systems*, vol 9, no. 3, March 1998.

[4] Arvo, J. & Heirich, A. “A Competitive Analysis of Load Balancing Strategies for Parallel Ray Tracing.” Kluwer Academic Publishers, Boston. California Institute of Technology, 1998.

[5] Fellner, D., Schafer, S., and Zens, M. “Photorealistic Rendering in Heterogeneous Networks,” *Advances in Parallel Computing*, vol. 12, pp. 113–120, 1998.

[6] Sahoo, B. “Dynamic Load Balancing Strategies in Heterogeneous System (Doctoral dissertation),” Retrieved from *ethesis.nitrkl.ac.in/5642/1/dlbmain.pdf*, 2013.

[7] Wang, H. et al. “An Innovate Dynamic Load Balancing Algorithm Based on Task Classification,” *IJACT: International Journal of Advancements in Computing Technology*, vol. 4, no. 6, pp. 244–254, 2012.

[8] Sidhu, M., Thulasiraman, P., and Thulasiram, R. “A Load-rebalance PSO Heuristic for Task Matching in Heterogeneous Computing Systems,” *IEEE Symposium on Swarm Intelligence (SIS)*, pp. 180–187, 2013.

Analog Realization of a Low-Voltage Sixteen Selectable Fractional-Order Differentiator in a 0.35um CMOS Technology

Geoffrey L. Abulencia and Alexander C. Abad

***Abstract*—This paper presents the design and implementation of an analog fractional-order differentiator (FOD) in a microelectronics scale. It focused on the design and implementation of sixteen selectable fractional-order (0.10, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85 and 0.90) differentiators in a 0.35um CMOS technology operated at 1.5-V supply. In comparison with the previous work that uses generic microcontroller for switching an FOD from one order to the next, this design of a 16 selectable FOD was realized in an analog microelectronic scale, thus, the physical implementation is relatively smaller. The authors employed reusability of resistors and capacitors when switching from one order to the other. The RC ladder in the design was implanted using NMOS capacitor and NWELL resistors while the IC design was implemented using TANNER software. The whole chip layout of the design has a dimension of 11.55mm x 8.32mm or equivalent to a final area of 96.10mm2. Each fractional order was characterized in terms of its frequency response —magnitude and phase response—in the bandwidth from 10Hz to 1kHz.**

***Index Terms*—constant phase element, resistor-capacitor ladder, selectable fractional-order differentiator, CMOS**

I. Introduction

THE concept of fractional calculus dates back to the time of Leibniz and L'Hospital in 1695 [1]. It is based on calculus with derivatives and integrals having non-integer orders. The concept fractional order calculus has not been easily adapted due to the complexity of its realization. Some definitions have been used for the general fractional calculus such as the Grunwald-Letnikov (GL) and the Riemann-Liouville (RL) [2]. Recently, it became a powerful and widely used tool for dynamical systems modeling [3], [4], [5], processes control [6], [7], [8], signal processing [9], [10], [11], and in many other fields of science and engineering.

Fractional-order differentiator (FOD) is a differentiator that performs non-integer-order differentiation, e.g. ½-order differentiation which is half derivative of a function. FOD can be realized through one of the following ways: a) Poly-Methyl-Methacrylate (PMMA) [12]; b) $LiN_2H_5SO_4$ or commonly known as the Lithium Salt [13]; c) Field Programmable Gate Arrays (FPGA) [14]; or d) electric component in the form of a Resistor-Capacitor (RC) ladder network [15], [16].

This paper focuses on the analog realization of a fractional-order differentiator implemented on a single integrated circuit (IC) design layout similar to the one presented in [17], [18].

II. Theory of Fractional-order Differentiator

A. Fractional Order Differentiator

A fractional-order differentiator is the generalization of a basic differentiator. An FOD can also be realized using operational amplifier (op-amp) circuit as shown in Figure 1. Instead of using a simple capacitor at its input side, an FOD uses a constant phase element (CPE).

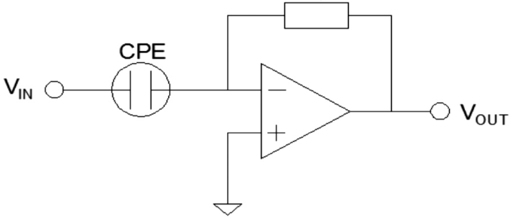


Fig. 1. Fractional-order differentiator circuit implementation

G. L. Abulencia and A. C. Abad are with the Electronics and Communications Engineering Department, Gokongwei College of Engineering of De La Salle University, Manila, Philippines (e-mail: jheof_abu07@yahoo.com)

According to [15], a fractional-order differentiator has a continuous-time transfer function of

$$G(s) = s^{\alpha} \tag{1}$$

wherein the magnitude of impedance depends on frequency according to the order of differentiation (α). Its value in decibels varies with the expression (20*α) dB/dec. Furthermore, the phase of the impedance is constant at the expression $(90*\alpha)^{o}$.

B. Constant Phase Element

An ideal constant phase element is composed of infinite number of lumped-sum parallel resistor-capacitor (RC) networks according to the concept of continued fraction expansions (CFE) representing an ideal transmission line [19]. Practically, the CPE in Figure 1 can be electronically realized using an RC ladder that approximates a fractional-order element with the schematic shown in Figure 2. This RC ladder was presented in [15] and was used successfully in the practical realization of fractional-order system.

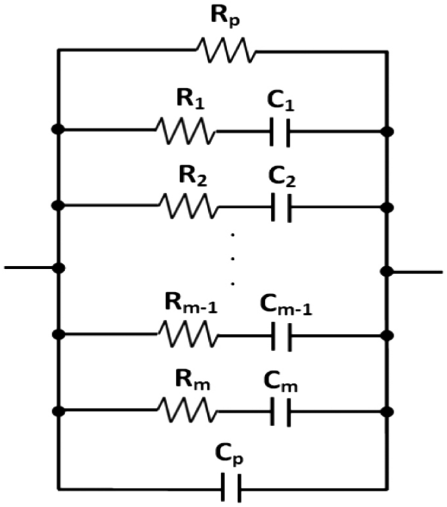


Fig. 2. CPE realization using RC ladder based on [15]

The following discussions of CPE was lifted from [15] and [18]. Generally, the higher the number of branches, the better the approximation of a CPE [15]. But for practical purposes, [15] and [18] only used five (5) branches in their CPE realization using RC-ladder network which is fit enough to meet the desired frequency band of interest. The method of RC ladder computation in this study are based primarily on [18] which is an improved and more general way compared to RC ladder network design presented in [16]. Table 1, which is based from [18], shows a summary comparing the original method of computation [16] and the optimized RC ladder branch values computation developed by the authors of this paper. The established optimization procedures developed by the authors are more general, straightforward and flexible since the recalibration step from the old method is eliminated. In effect, infinite sets of RC ladder can be obtained using the new approach which allows common values of resistor and capacitor for different fractional orders.

In the new CPE design procedure, four initial values are needed: 1) the maximum allowable phase ripple (Δφ); 2) the number of RC ladder branches (m); 3) the order of differentiation (α); and 4) the initial value of R_1 (same for all orders). The remaining nine values of resistors (R2 to R5) and capacitors (C1 to C5) for the ladder branch can be computed using equations (2) to (7) as follows:

TABLE I
COMPARISON OF THE ORIGINAL AND OPTIMIZED RC LADDER BRANCH VALUES COMPUTATION

STEPS	ORIGINAL COMPUTATION [15]	OPTIMIZED COMPUTATION
Initial values needed	Phase Ripple (Δϕ) Desired Gain (Dr) Order (α) No. of branch (m) Initial R1 and C1	Phase Ripple (Δϕ) Order (α) No. of branch (m) Initial R1
Determination of parameters ‘a’ and ‘b’	$ab \approx \frac{0.24}{1+\Delta\varphi}$ $\log a = \alpha \log(ab)$	$ab = \frac{0.24}{1+\Delta\varphi}$ $\log a = \alpha \log(ab)$
Determination of RC ladder branch values	$R_k = R_1 a^{k-1}$ $C_k = C_1 b^{k-1}$	$R_k = R_1 a^{k-1}$ $C_k = \frac{b^{k-1}}{100R_1}$
Determination of ‘Rp’ and ‘Cp’	$R_p = R_1 \frac{1-a}{a}$ $C_p = C_1 \frac{b^m}{1-b}$	$R_p = R_1 \frac{1-a}{a}$ $C_p = C_1 \frac{b^m}{100R_1(1-b)}$
Approximation of the minimum and maximum frequencies of operation	$\omega_{max} \approx \frac{\omega_{min}}{(ab)^m}$ $\omega_{av} = \sqrt{\omega_{max}\omega_{min}}$	$\omega_{max} \approx \frac{\omega_{min}}{(ab)^m}$ $\omega_{av} = \sqrt{\omega_{max}\omega_{min}}$
Recalibration of the resistor and capacitor values	$Y(j\omega) = \frac{1}{R_p} + j\omega C_p \ldots$ $\ldots + \sum_{k=1}^{m} \frac{j\omega C_k}{1 + j\omega R_k C_k}$ $D = \frac{1}{\lvert Y(j\omega_{av})\rvert \omega_{av}^{\alpha}}$ **Resistors multiplied by Dr/D** **Capacitors divided by Dr/D**	NO RECALIBRATION NEEDED

$$R_k = R_1 a^{k-1} \tag{2}$$

and

$$C_k = \frac{b^{k-1}}{100 R_1} \tag{3}$$

According to [15], the parameters ‘a’ and ‘b’ ranges between 0 and 1 which can be computed using their relationships with the order of differentiation (α) and the maximum allowable phase ripple (Δφ) as

$$\Delta\varphi = \frac{0.24}{ab} - 1 \tag{4}$$

and

$$\alpha = \frac{\log a}{\log (ab)} \tag{5}$$

To replace the truncated sections by a simple network for the CPE, the resistive side of the ladder (upper portion) can be represented by a single resistor R_p while the capacitive (lower portion) can be represented by a single capacitor C_p with values computed using

$$R_p = R_1 \frac{1-a}{a} \tag{6}$$

and

$$C_p = \frac{b^m}{100 R_1 (1-b)} \tag{7}$$

For this study, the authors used phase ripple Δφ = 0.2, m = 5, R_1 = 200kΩ. The values for the sixteen fractional orders are α = ***0.10, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85,*** and ***0.90.*** Table 2 shows the list of all the computed resistor and capacitor values for the sixteen fractional orders.

TABLE II

List of All Resistor and Capacitor Values for 16FOD RC Ladder Network

ORDER	**0.10**	**0.20**	**0.25**	**0.30**	**0.35**	**0.40**	**0.45**	**0.50**
R1(Ω)	200000.00	200000.00	200000.00	200000.00	200000.00	200000.00	200000.00	200000.00
R2(Ω)	170267.98	144955.93	133748.06	123406.77	113865.06	105061.11	96937.87	89442.72
R3(Ω)	144955.93	105061.11	89442.72	76146.16	64826.26	55189.19	46984.76	40000.00
R4(Ω)	123406.77	76146.16	59813.95	46984.76	36907.23	28991.19	22773.01	17888.54
R5(Ω)	105061.11	55189.19	40000.00	28991.19	21012.22	15229.23	11037.84	8000.00
Rp(Ω)	34923.79	75945.93	99069.76	124131.32	151293.00	180730.79	212635.41	247213.60
ORDER	**0.55**	**0.60**	**0.65**	**0.70**	**0.75**	**0.80**	**0.85**	**0.90**
R1(Ω)	200000.00	200000.00	200000.00	200000.00	200000.00	200000.00	200000.00	200000.00
R2(Ω)	82527.08	76146.16	70258.60	64826.26	59813.95	55189.19	50922.00	46984.76
R3(Ω)	34053.60	28991.19	24681.35	21012.22	17888.54	15229.23	12965.25	11037.84
R4(Ω)	14051.72	11037.84	8670.39	6810.72	5349.92	4202.44	3301.08	2593.05
R5(Ω)	5798.24	4202.44	3045.85	2207.57	1600.00	1159.65	840.49	609.17
Rp(Ω)	284689.37	325305.56	369325.32	417033.86	468740.30	524779.66	585515.03	651339.92
ORDER	**0.10**	**0.20**	**0.25**	**0.30**	**0.35**	**0.40**	**0.45**	**0.50**
C1(F)	5.0000E-08	5.0000E-08	5.0000E-08	5.0000E-08	5.0000E-08	5.0000E-08	5.0000E-08	5.0000E-08
C2(F)	1.1746E-08	1.3797E-08	1.4953E-08	1.6207E-08	1.7565E-08	1.9037E-08	2.0632E-08	2.2361E-08
C3(F)	2.7595E-09	3.8073E-09	4.4721E-09	5.2531E-09	6.1703E-09	7.2478E-09	8.5134E-09	1.0000E-08
C4(F)	6.4826E-10	1.0506E-09	1.3375E-09	1.7027E-09	2.1676E-09	2.7595E-09	3.5129E-09	4.4721E-09
C5(F)	1.5229E-10	2.8991E-10	4.0000E-10	5.5189E-10	7.6146E-10	1.0506E-09	1.4496E-09	2.0000E-09
Cp(F)	4.6763E-11	1.1049E-10	1.7067E-10	2.6467E-10	4.1235E-10	6.4592E-10	1.0183E-09	1.6180E-09
ORDER	**0.55**	**0.60**	**0.65**	**0.70**	**0.75**	**0.80**	**0.85**	**0.90**
C1(F)	5.0000E-08	5.0000E-08	5.0000E-08	5.0000E-08	5.0000E-08	5.0000E-08	5.0000E-08	5.0000E-08
C2(F)	2.4234E-08	2.6265E-08	2.8466E-08	3.0852E-08	3.3437E-08	3.6239E-08	3.9276E-08	4.2567E-08
C3(F)	1.1746E-08	1.3797E-08	1.6207E-08	1.9037E-08	2.2361E-08	2.6265E-08	3.0852E-08	3.6239E-08
C4(F)	5.6933E-09	7.2478E-09	9.2268E-09	1.1746E-08	1.4953E-08	1.9037E-08	2.4234E-08	3.0852E-08
C5(F)	2.7595E-09	3.8073E-09	5.2531E-09	7.2478E-09	1.0000E-08	1.3797E-08	1.9037E-08	2.6265E-08
Cp(F)	2.5955E-09	4.2132E-09	6.9442E-09	1.1678E-08	2.0188E-08	3.6335E-08	6.9718E-08	1.5041E-07

III. Conceptual Design of a Selectable FOD

A conceptual selectable fractional-order differentiator was shown in [16] using discrete components. The conceptual design's order of differentiation can be chosen from 0 to 1 with an increment of 0.05.

The design utilizes cascaded operational amplifiers and resistor-capacitor ladders as its main components, while a generic microcontroller is introduced for switching purposes. Initial simulation results through Matlab and LTSpiceIV show that the designed resistor-capacitor ladders can perform as an analog FOD [16].

For the sole purpose of switching circuitry, utilization of microcontroller apparently results to much lower component density in the hardware implementation. While the trend in electronics physical realization is to go smaller and lightweight, the authors were inspired to design and implement a sixteen selectable FOD in microelectronic scale using 0.35um CMOS technology. The whole design was implemented in a relatively much simpler architecture wherein the switching circuitry is already an integrated part of the whole circuit eliminating the use of microcontroller.

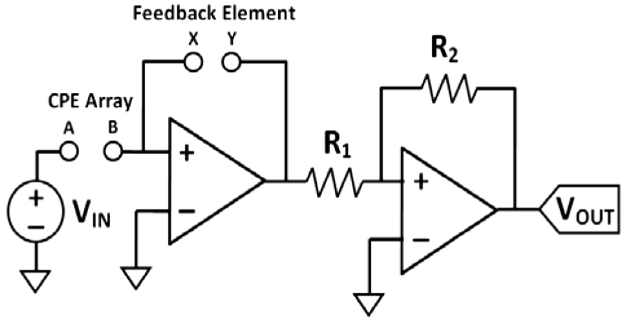


Fig. 3. Schematic layout of a selectable FOD utilizing selector module circuitry

Figure 3 shows the basic schematic overview of a selectable FOD. Basically, change of α would require a change of ladder values (see Table 2) as well as the value of feedback resistor (see Table 3). There will be two sets of selector module circuit: one for CPE module (pins A and B) and the other for feedback (pins X and Y). Table 3 also shows the magnitude gain of the FOD at different frequencies.

Reuse of resistors and capacitors was adopted in the design to further scale down the physical dimension of the implementation. This simply means that some of the RC components in one fractional order to another are being utilized as switching occurs. As can be observed in Table 2, all fractional orders have 200-kΩ R_1 and 50-nF C_1. Instead of having sixteen 200-kΩ R_1 and sixteen 50-nF C_1 in the circuit design, it is possible to have just one 200-kΩ R_1 and one 50-nF C_1 for all FODs. The concept of reusability was initially employed by the authors in two-order FOD (2FOD) design [18]. Generally, 16FOD is just an expansion of 2FOD.

TABLE III
Magnitude Gain and Corresponding R_F for 16FOD

ORDER	Magnitude Gain of the FOD (in dB)			Feedback Resistor (R_F) in Ω
	10Hz	100Hz	1kHz	
0.10	3.60	5.60	7.60	17k
0.20	7.19	11.19	15.19	51k
0.25	8.99	13.99	18.99	77k
0.30	10.79	16.79	22.79	110k
0.35	12.59	19.59	26.59	155k
0.40	14.39	22.39	30.39	206k
0.45	16.18	25.18	34.18	271k
0.50	17.98	27.98	37.98	351k
0.55	19.78	30.78	41.78	436k
0.60	21.58	33.58	45.58	531k
0.65	23.38	36.38	49.38	646k
0.70	25.17	39.17	53.17	716k
0.75	26.97	41.97	56.97	766k
0.80	28.77	44.77	60.77	804k
0.85	30.57	47.57	64.57	814k
0.90	32.37	50.37	68.37	816k

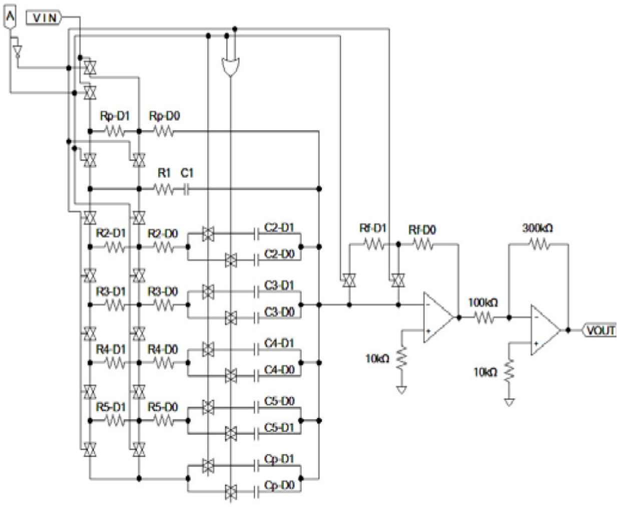


Fig. 4. Top-level schematic of a 2FOD [18]

An understanding of reusability can be simply deduced from Figure 4 which shows the top-level schematic of a two-order selectable FOD. The control bit A determines which order of differentiation to actuate: A="0" for FOD(0.25) and A="1" for FOD(0.50). The design utilizes several transmission gates to switch from one order to the next. An OR gate is used for actuating set of capacitors. For both orders of differentiation, all capacitor D0's have to be activated. Capacitors D1's connected in parallel to capacitor D0's supplement the necessary capacitance values. The switch-activated RC ladder circuit serves as the input impedance to the two-stage CMOS operational amplifier that is then post-cascaded to another inverting amplifier [18].

Ideally, the magnitude of an FOD is 0dB at ω=1 rad/s. According to [16], the average frequency ω_{av} can be computed using the equation shown in Table 2. For instance, if the order of differentiation is 0.25, then the average frequency is around 5590.2 rad/s or equivalent to 890Hz. Using the closest decade point, which is at 1kHz, the magnitude of the gain should be at $|G(j\omega)|_{\omega=2\pi(1000)} = \omega^{0.25}|_{\omega=2\pi(1000)} = 8.9032$ which is around 18.99dB. Table 3 summarizes the magnitude gain for every decade from 10Hz to 1kHz, as well as the corresponding feedback resistor values empirically chosen for sixteen fractional orders of differentiation.

According to [18], the concept of reusability can be applied for other RC ladder branches. The magnitude of R_2 for FOD(0.50) from Table 2 can be reused and add up with 44305.34Ω to complete the 133748.06-Ω R_2 needed for FOD(0.25). Likewise, with the capacitor, the magnitude of C_2 for FOD(0.25) can be reused and add up with 7.4072nF to accummulate a total of 22.361-nF C_2 needed for FOD(0.50). This scheme significantly reduces the overall dimension of the analog realization since duplication of resistor and capacitor values is averted. Strategic placing of the transmission gates must be thoroughly taken into consideration to optimize reusability.

Figure 5 shows the top-level schematic of 16FOD which consists of a 4-bit parallel-in parallel-out (PIPO) register, a 4x16 line decoder, arrays of CPE elements, arrays of feedback resistor and cascaded inverting amplifiers. The design of 16FOD is very similar to anexpanded Figure 4. Resistor array and capacitor array magnification of CPE elements are shown in Figures 6 and 7 respectively after employing reusability for the sixteen fractional orders.

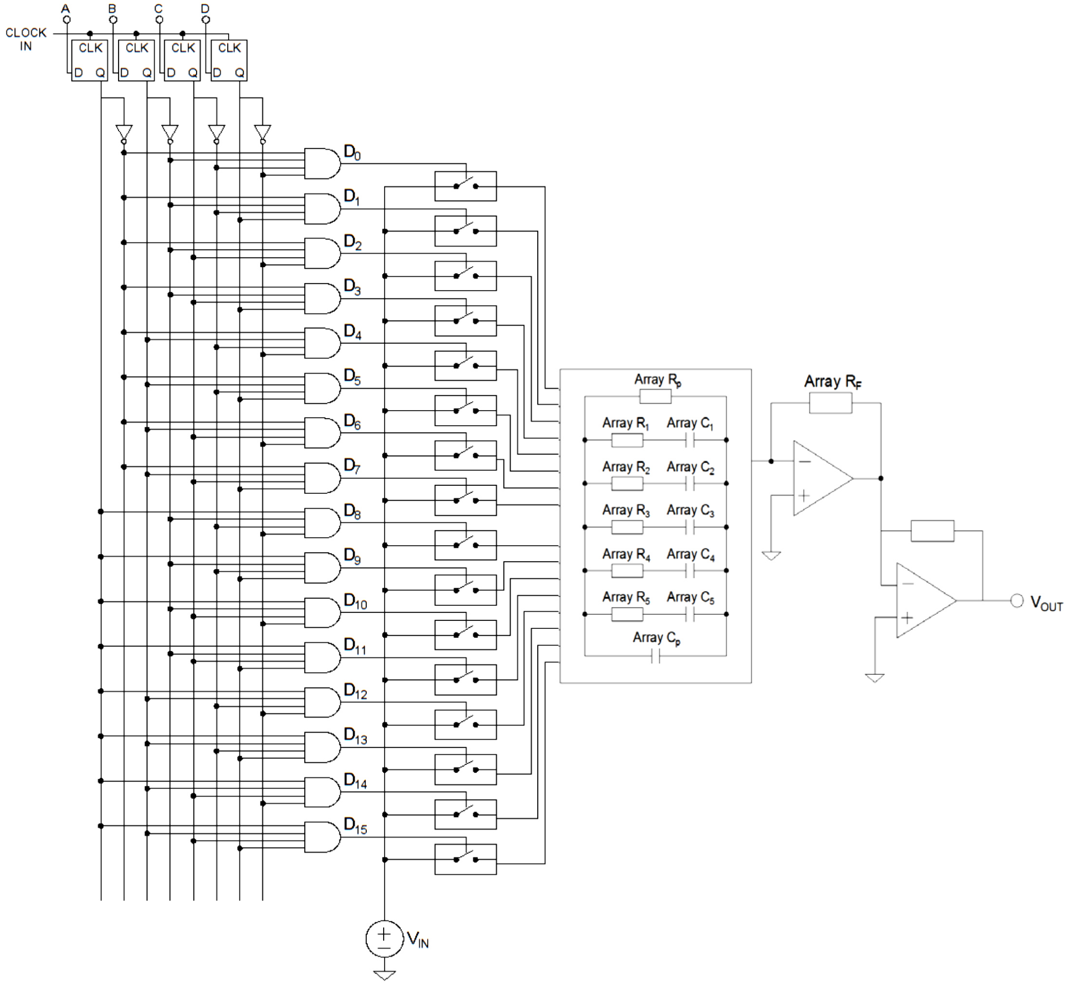


Fig. 5. Top-level schematic of a sixteen selectable FOD

The 4-bit PIPO register catches the input order of differentiation. If for instance all bits (A, B, C, D) are “low,” FOD(0.10) is activated and if all bits are “high,” FOD(0.90) is activated. Transmission gates in the RC arrays are all normally open. The 4x16 line decoder that signals which set of transmission gates to trigger, in essence, dictate which series-connected resistors and parallel-connected capacitors to add up. This scheme also applies for the array of feedback resistors. The output of each resistor array branch is cascaded to its corresponding capacitor array to generate the necessary input impedance for the inverting operational amplifier.

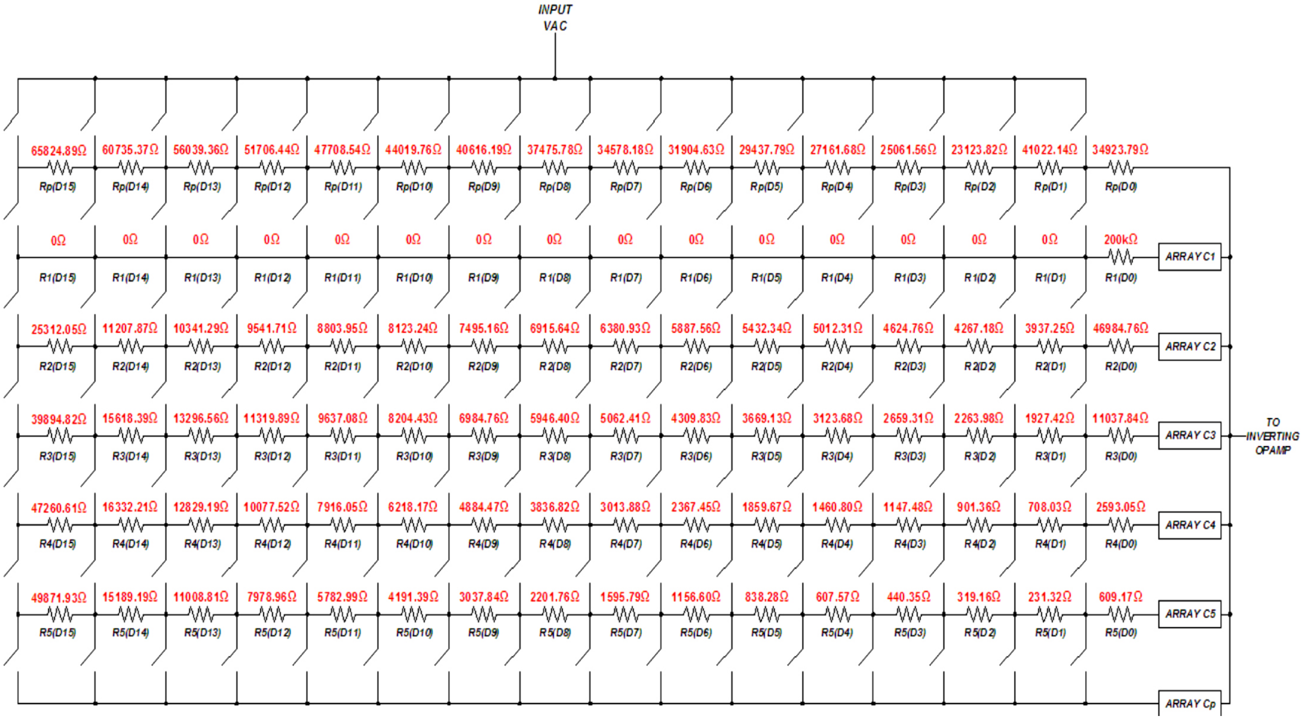


Fig. 6. Resistor array of CPE employing reusability for selectable 16FOD

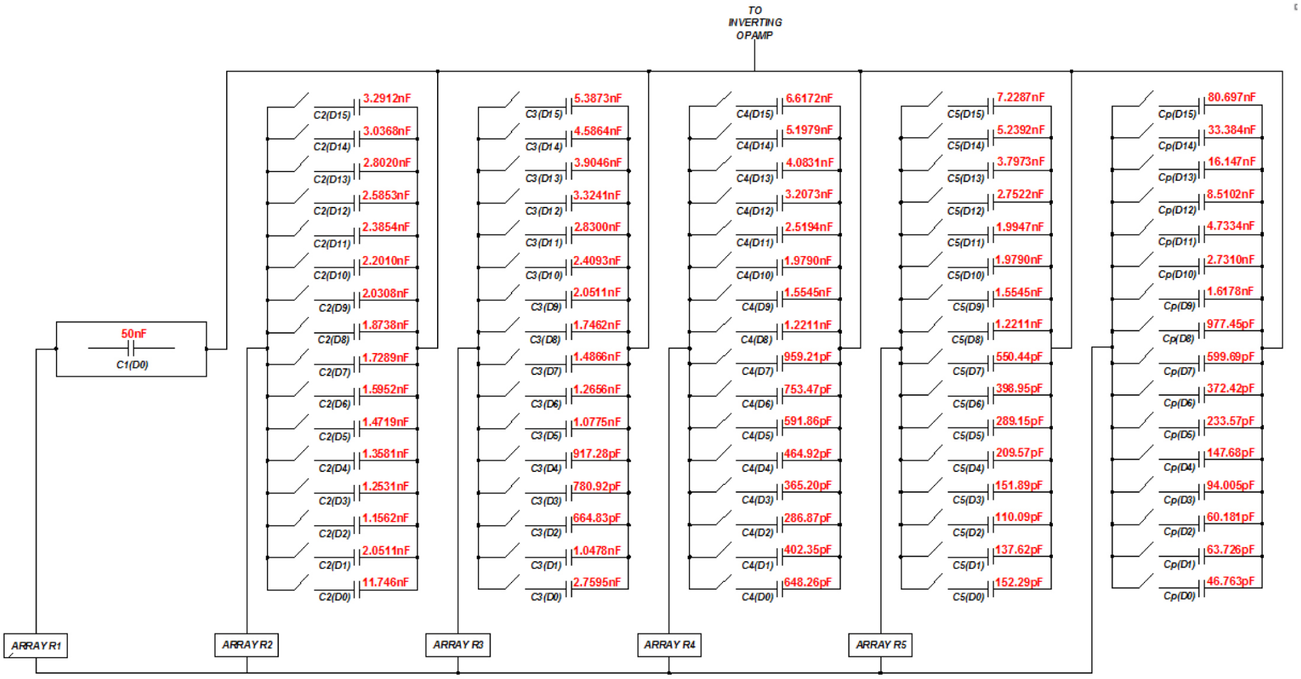


Fig. 7. Capacitor array of CPE employing reusability for selectable 16FOD

IV. Integrated Circuit (IC) Layout Implementation

Like the layout designs in [17] and [18], Tanner Software was also used to produce a single-chip layout of the whole circuit in this study. L-edit tanner tool was used as the layout editor.

A. Operational Amplifier CMOS Module

The op-amp used in [18], [20], [21] shown in Figure 5 was adopted for this study. The design works at low voltage supply of around 1.5 volts instead of the typical 3.3V for 0.35um CMOS technology. The adopted op-amp can be operated either by using unipolar or bipolar supply. It was designed to operate at +/-0.75V or 1.5V to GND. Either way, the op-amp exhibits same parametric response.

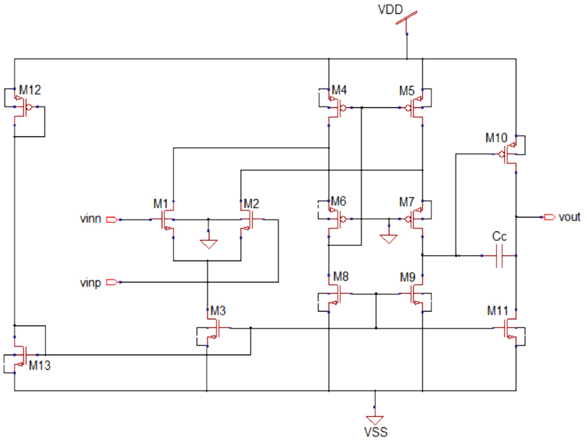


Fig. 8. Op-amp low voltage design [40]

Ideally, the voltage gain phase of the op-amp at 0dB should be at 45° and at least a gain of 60dB should be achieved. It is likewise important to ensure that all transistors are operating in saturation region. Step by step computation for the transistor sizes were guided by [20], [21], [22]. Table 4 shows the summary of the computed width size of transistors M1 to M11 alongside with its adjusted width size (actual size). A bias voltage of 0.65V is needed to operate the op-amp with its desired functionality. This can be achieved through the use of bias transistors M12 and M13 having values equal to 6um and 1um, respectively. All transistors have the length of 1um.

B. Experimental Validation Setup

The actual layout underwent physical verification processes through Design Rule Check (DRC), ensuring that the created mask layout conform to the complex set of design rules and Layout Verification Schematic (LVS) assuring it represents the circuit desired to be fabricated.

TABLE IV
Op-Amp Transistor Sizes Used for the Selectable FOD

Transistor	Computed Width Size (um)	Actual Width Size (um)
M1	12.15	17
M2	12.15	17
M3	24.30	16
M4	22.73	22
M5	22.73	22
M6	22.73	18
M7	22.73	18
M8	11.54	20
M9	11.54	20
M10	27.27	28
M11	11.5385	37

Parasitic capacitances were considered in the design to create an accurate analog model of the circuit. However, only nodal parasitic capacitance effects were counted in the simulation as this was the only option possible in the simulation software. Parasitic resistances and inductances were not included in the extraction. All nodal parasitic capacitances were considered since all capacitance less than 0 femtofarad were set to be ignored.

The ideal magnitude and phase response of an FOD for different orders are listed in Table 5, which was used to validate the frequency response. In reference to [16] and [18], ideal responses, specifically the phase response, are not attainable for the whole frequency band due to the gain-bandwidth limitation of the op-amp.

C. CMOS Layout View of a Selectable FOD

Shown in Figure 9 is the physical layout implementation of the 16FOD. It has an overall IC dimension of 11.55mm x 8.32mm. The total area of 16FOD is equivalent to just three times the layout of semi-differentiator presented in [17].

As shown in Figure 10, the microelectronic-scaled selectable FOD exhibits magnitude response that is almost equal to the ideal gain. On the other hand, the phase response significantly deviates from the ideal as the frequency increases. Table 5 presents the post-layout results of gain and phase response for the sixteen fractional orders. Not all FODs are working for this whole frequency band of interest. Theoretically, the RC ladder can perform the desired constant phase element. However, when incorporated in an op-amp circuit, the range of frequencies diminishes. This is mainly due to the characteristics and gain-bandwidth-product (GBP) limitation of the op-amp. For higher bandwidth applications, a design of op-amp with higher GBP is necessary.

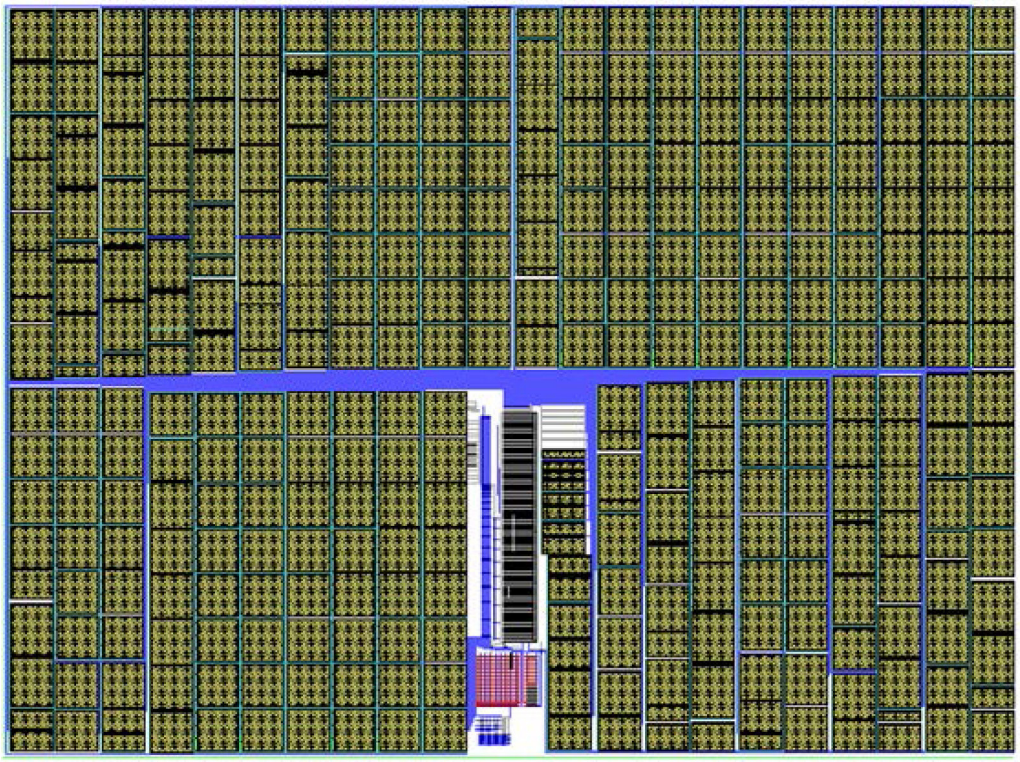


Fig. 9. Top-level layout of a selectable 16FOD

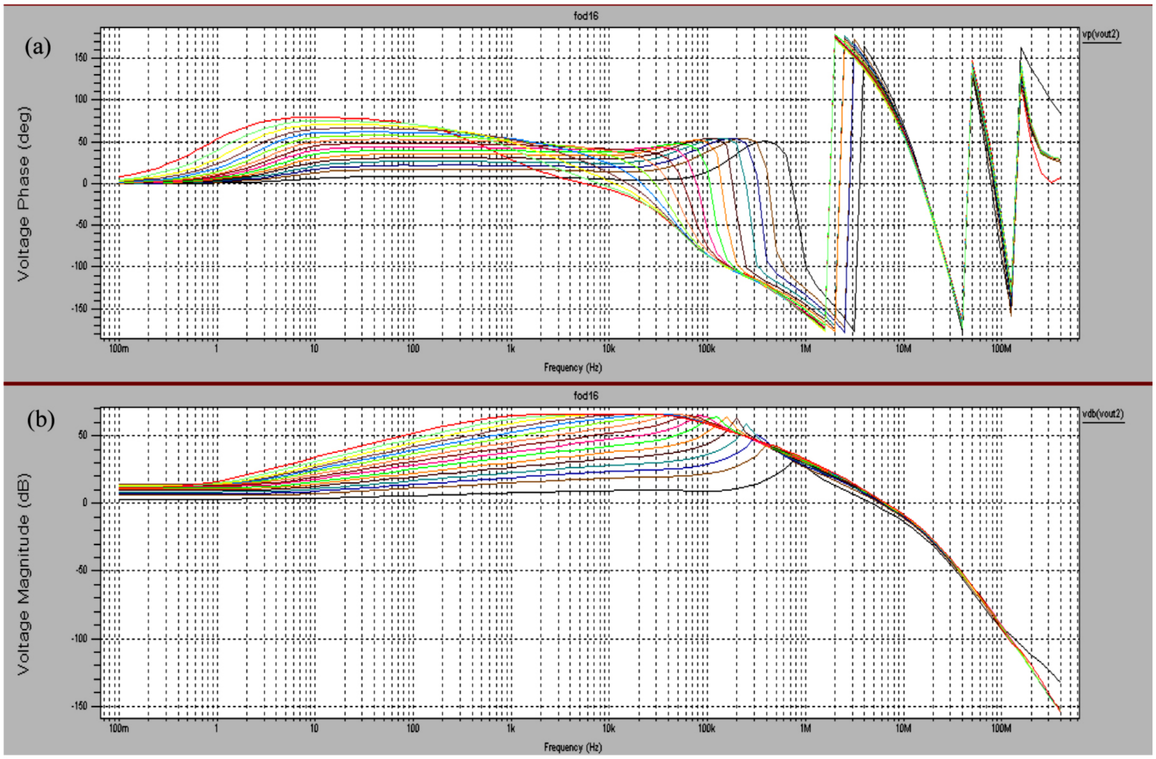


Fig. 10. Frequency response of the designed 16FOD for: a) phase response and b) magnitude response

TABLE V

6FOD Gain and Phase Response

ORDER	FREQ	IDEAL GAIN (dB)	Post-Layout Results (dB)	IDEAL PHASE (degrees)	Post-layout Result (degrees)
FOD(0.10)	10Hz	3.60	3.83	9.0	6.02
	100Hz	5.60	5.66		8.77
	1kHz	7.60	7.58		8.45
FOD(0.20)	10Hz	7.19	7.33	18.0	13.02
	100Hz	11.19	11.52		19.30
	1kHz	15.19	15.21		17.06
FOD(0.25)	10Hz	8.99	9.08	22.5	17.01
	100Hz	13.99	14.26		23.80
	1kHz	18.99	19.01		20.13
FOD(0.30)	10Hz	10.79	10.73	27.0	20.75
	100Hz	16.79	16.95		28.21
	1kHz	22.79	22.63		24.27
FOD(0.35)	10Hz	12.59	12.42	31.5	25.10
	100Hz	19.59	20.01		32.56
	1kHz	26.59	26.46		28.84
FOD(0.40)	10Hz	14.39	14.18	36.0	29.85
	100Hz	22.39	22.60		36.55
	1kHz	30.39	30.13		33.18
FOD(0.45)	10Hz	16.18	15.91	40.5	34.40
	100Hz	25.18	25.54		41.04
	1kHz	34.18	33.83		37.70
FOD(0.50)	10Hz	17.98	17.99	45.0	39.29
	100Hz	27.98	28.13		45.45
	1kHz	37.98	37.68		41.28
FOD(0.55)	10Hz	19.78	19.64	49.5	44.48
	100Hz	30.78	31.10		49.65
	1kHz	41.78	41.31		44.87
FOD(0.60)	10Hz	21.58	21.35	54.0	49.36
	100Hz	33.58	33.58		52.60
	1kHz	45.58	45.09		48.41
FOD(0.65)	10Hz	23.38	23.40	58.5	54.36
	100Hz	36.38	36.69		56.61
	1kHz	49.38	49.14		50.74
FOD(0.70)	10Hz	25.17	24.84	63.0	59.50
	100Hz	39.17	39.09		60.63
	1kHz	53.17	52.51		52.01
FOD(0.75)	10Hz	26.97	26.58	67.5	65.02
	100Hz	41.97	41.55		64.64
	1kHz	56.97	55.62		51.69
FOD(0.80)	10Hz	28.77	28.31	72.0	70.19
	100Hz	44.77	44.51		67.91
	1kHz	60.77	59.22		48.64
FOD(0.85)	10Hz	30.57	30.76	76.5	74.95
	100Hz	47.57	47.23		70.51
	1kHz	64.57	62.10		41.63
FOD(0.90)	10Hz	32.37	33.95	81.0	79.31
	100Hz	50.37	51.50		71.42
	1kHz	68.37	64.69		29.52

Transient response of the design was also tested and analyzed. Sinusoidal input signal was used to clearly show the delay or the phase shift between the input and output signal. Transient response of FOD(0.25) is shown in Figure 11 for an input signal, where the frequency was set to 60Hz. At 60-Hz frequency, the magnitude gain of the FOD should be at $|G(j\omega)|_{\omega=2\pi f} = \omega^{0.25}|_{\omega=2\pi(60)} = 4.41$. Using the computed gain, with an input signal amplitude of 5mV, the output signal amplitude should then be equal to 22.03mV. This is close to the graph showing an output peak voltage of 22.16mV. Meanwhile, the time delay between two sine waves is equal to 1.03ms. The phase shift of the output signal with respect to the input signal is equal to 22.25°, which is close to the ideal phase angle for FOD(0.25) of 22.5°.

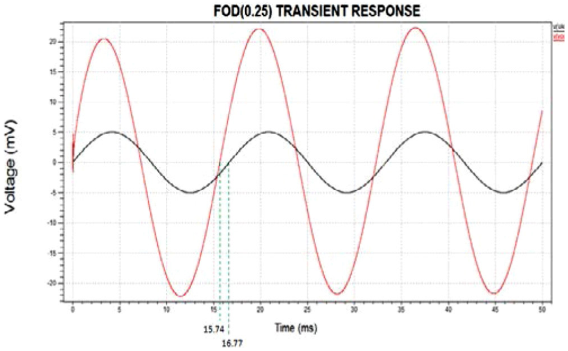


Fig. 11. FOD(0.25) transient response

V. Other Design Considerations

Initial value of R_1 was carefully assessed. Using the computations in Table 1, R_1 value must be high to have capacitor values relatively smaller. Figure 8 shows that almost 95% of the total chip area is consumed by capacitors. However, it was also taken into consideration that other capacitance in the RC ladder must not fall down to tens of picofarad range, which is foreseen to be highly sensitive to parasitic capacitances.

The type of the capacitor implemented in this design was carefully chosen. NMOS-type of capacitor was considered in the study since it has higher capacitance value compared to poly-to-poly capacitor. Shown in Figure 12 is the comparison between the two types of capacitor. With regards to resistor, a poly type resistor was chosen for this study.

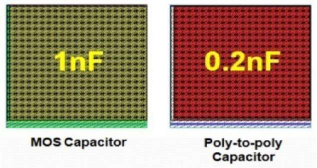


Fig. 12. Comparison of capacitance value of MOS and poly-to-poly capacitor having same physical dimension

T-gates characterization was done to examine the parasitic effects of the switches to the overall design. The number of t-gates in parallel has a significant effect to the phase response of an FOD. As a result, a 4-TG switch was used in the design to give a better phase response with the least physical dimension layout of the switch as possible. Shown in Figure 13 is the result of t-gate characterization.

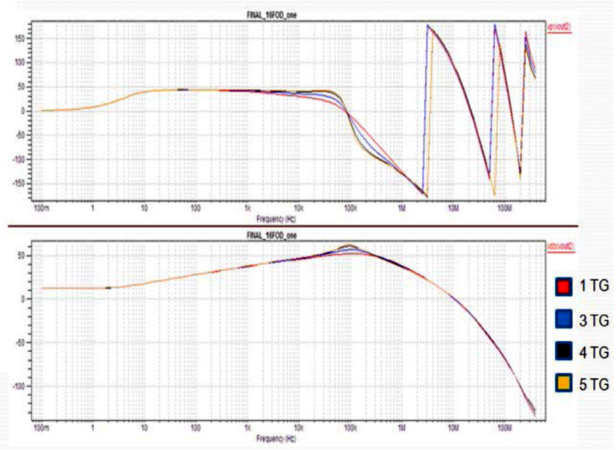


Fig. 13. Phase and gain response of an FOD for a switch with increasing number of transmission gate (TG)

VI. Conclusions

In this study, a design of a low-voltage selectable sixteen fractional-order differentiator (***0.10, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, and 0.90)*** has been designed and implemented in a microelectronic scale using 0.35um technology. Unlike in [16] which uses microcontroller for switching purposes, this design is successfully realized in an analog microelectronic scale, and thus, relatively smaller. The design employed reusability of capacitors and resistors when switching from one order to another. The final physical layout of the design using L-Edit has a dimension of 11.55mm x 8.32mm or equivalent to 96.10mm^2, which is just about three times the area of a semi-differentiator in [17]. The whole chip was powered using 1.5 Volt supply. Several design considerations such as the type of capacitor and resistor to implement, transmission gate design, and the initial value of R_1 were evaluated. The overall design was characterized in its frequency response —the magnitude and phase response for every order. The gain-bandwidth limitation of the op-amp actually bounds the frequency response of the 16FOD which opens up for possible research study in the future.

Acknowledgment

The authors would like to thank the Engineering Research and Development for Technology (ERDT) of the Department of Science and Technology (DOST) – Philippines for funding this research, as well as to the De La Salle University – Manila research critics especially to Dr. Emmanuel Gonzalez and Engr. Roderick Yap for extending their profound knowledge towards the realization of this study.

References

[1] IB. Ross, "The development of fractional calculus 1695–1900," *Historia Mathematica*, vol. 4, iss. 1, 1977, pp 75–89.

[2] Gonzales, E. (2013). *Design of robust fractional-order control systems.* Dissertation, PhD in ECE, De La Salle University, Manila, Philippines.

[3] I. Petras, D. Sierociuk, and I. Podlubny, "Identification of parameters of a half-order system," *IEEE Transactions on Signal Process.,* vol. 60, iss. 10, 2012, pp. 5561–5566.

[4] L. Dorčák, J. Valsa, J. Terpák, P. Horovčák, and E. Gonzalez, "Modeling and identification of fractional-order dynamical systems," *Proceedings of the 11th International Multidisciplinary Scientific Geoconference (SGEM2011)*, Jun. 20–25, 2011, Bulgaria, pp. 553–560.

[5] L. Dorčák, "Numerical models for simulation of the fractional-order control systems," The Academy of Sciences, Institute of Experimental Physics, UEF-04-94, Technical University of Koˇsice, Slovakia.

[6] M. Axtell and M. E. Bise, "Fractional calculus applications in control systems," *Proceedings of the IEEE 1990 National Aerospace and Electronics Conference*, May 21–25, 1990, New York, pp. 563–566.

[7] R. S. Barbosa, J. A. T. Machado, and I. M. Ferreira, "PID controller tuning using fractional calculus concepts," *Fractional Calculus & Applied Analysis*, vol. 7, no. 2, 2004, pp. 119–34.

[8] J. A. T. Machado, "Analysis and design of fractional-order digital control systems," *Journal of Systems Analysis, Modeling and Simulation*, vol. 27, 1997, pp. 107–122.

[9] R. Magin, M. Ortigueira, I. Podlubny, and J. Trujillo, "On the fractional signals and systems," *Signal Processing*, vol. 91, iss. 3, Mar. 2011, pp. 350–371.

[10] W.M. Ahmad and K. Assaleh, "Modeling of speech signals using fractional calculus," *9th International Symposium on Signal Processing and Its Applications*, 2007, pp. 1–4.

[11] A. Oustaloup, "Fractional order sinusoidal oscillators: optimization and their use in highly linear FM modulation," *IEEE Transactions on Circuits and Systems*, vol. CAS-28, no. 10, Oct. 1981.

[12] Biswas, K., Sen, S., and Dutta, P.K. (2006). Realization of a coznstant phase element and its performance study in a differentiator circuit. IEEE Transactions on Circuits and Systems, 53 (9), pp. 802–806.

[13] Sheng, H., Sun, H., Coopmans, C., Chen, Y., and Bohannan, G. (2010). Physical experimental study of variable-order fractional integrator and differentiator. IFAC Workshop Fractional Differentiation and Its Applications, 4, pp 1–6.

[14] Jiang, C.X., Carletta, J.E., and Hartley, T.T. (2007). Implementation of fractional-order operators on field programmable gate arrays. Advances in Fractional Calculus, pp. 333–346.

[15] Valsa, J., Dvořák, P., and Friedl, M. (2011). Network model of the CPE.Radio Engineering, 20 (3), pp. 619–626.

[16] Gonzalez, E.A., Dorčák, L., Monje, C.A., Petráš, I., and Valsa, J., Caluyo, F.S., Gonzalez, R.M., Jundez, C.P.M., Reyes, D.D., & Villanueva, T.M.G. (2014). Conceptual design of a selectable fractional-order differentiator for industrial applications. Fractional Calculus and Applied Analysis, 17 (3), pp. 697–716.

[17] A. C. Abad, E. A. Gonzalez, R. Y. Yap, and L. Dorcak (2014, January 02). Analog realization of a fractional-order element on 0.35 um CMOS technology [Online]. http://vixra.org/pdf/1401.0016v1.pdf

[18] Abulencia, G. L. and Abad, A. C., Analog realization of a low-voltage two-order selectable fractional-order differentiator in a 0.35um CMOS technology, International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), Cebu, Philippines, 2015

[19] Podlubny, I., Petráš, I., Dorčák, L., Vinagre, B.M., and O'Leary, P. (2002). Analogue realizations of fractional-order controllers. Nonlinear Dynamics, 29, pp. 281–296.

[20] R. Yap and W. Y. Chung, "A 1.2V Reduced Output Reference Voltage Bandgap," Proceedings of the 10th DLSU Science & Technology Congress, July 23, 2008, De La Salle University–Manila.

[21] Yap, Roderick, "A Low Voltage Dynamic Power Saving Pulse Frequency Modulated Boost Converter Design for Driving a White LED", MS Thesis Document, Chung Yuan Christian University. Taiwan, January 2009.

[22] Allen, P. and Holberg, D. (2002). CMOS analog circuit design. New York: Oxford University Press.

Comparison of Logit and Neural Network Models in Inter-Island Discrete Choice Analysis

Krister Ian Daniel Roquel and Alexis Fillone

Abstract—**Logit-based models have often been used for discrete choice analysis. However, conventional logit models preserve a linear relationship that requires variables that are independent of each other, which is generally not the proper assumption. In this paper, the researcher addresses the non-linear behavior and inter-dependence of variables using neural networks in modeling inter-island travel choice. Neural network analysis was employed to a previous work to test the applicability of neural network in discrete choice models for inter-island travel. It was found that the neural network model is statistically acceptable in describing travel choice behavior, while the logit model is more inclined to model the decision making process. Also, it was found that the neural network model is capable of accurately predicting the minority, which has long been a problem when using logit models as these are usually treated as errors.**

Index Terms—**discrete choice, multinomial logit, neural network, inter-island travel**

I. Introduction

LOGIT-BASED models have often been used for discrete choice analysis. These are based on the random utility theory, which employs an abstract measurement of the degree of satisfaction for any choice an individual makes, with the assumption that rational people act to maximize their utility. However, conventional logit models preserve a linear relationship that requires variables that are independent of each other, which is generally not the proper assumption. In this paper, the non-linear behavior and inter-dependence of variables are addressed using neural networks in modeling inter-island travel choice.

Previous works on the application of neural networks on discrete choice behavior have shown potentials and advantages of employing neural networks over the traditional logit models. As early as the late 1990s, Nijkamp, et al. [1] conducted a study on the comparison of neural network and logit analysis in modeling inter-urban transport flows. Bentz and Merunka [2], Hensher and Ton [3], Cantarella and Luca [4], Vythoulkas and Koutsopoulos [5], Norets [6], Nakayama, et al. [7], and Dia [8] all have contributions on the field with their respective researches on using artificial neural networks on discrete choice applications. Even until recently, Pulugurta, et al. [9] still conducts studies on the comparison of the models developed using various approaches.

As choice decisions usually involve approximations that are not precisely captured by logit models, neural network models would always have a place in discrete choice analysis due to their capability of function inference based on observations. The latter does not need any prior knowledge of the characteristics of the variables and can account for wnon-linearity, which makes for an easier and more convenient model development process. In this paper, neural network analysis is employed to a previous work [10], to test the applicability of neural network in discrete choice models for inter-island travel.

II. Study Area

The data used in the study were gathered from terminals serving the inter-island network in the heart of the Visayan region in the Philippines. Major contributors to inter-island traffic in the region are the provinces of Iloilo and Negros Occidental, which are two highly urbanized provinces with populations of over 2.2 M and 2.9 M, respectively (NSO, 2009). Fig. 1 shows the inter-island travel options currently available to the public. As shown, inter-island travel can be done in four ways (A, B, C, and D) in this travel network.

K. I. D. Roquel is a graduate student of the Civil Engineering Department, De La Salle University, Manila, Philippines (e-mail: krister.roquel@dlsu.edu.ph)

A. Fillone is with the Civil Engineering Department, De La Salle University, Manila, Philippines

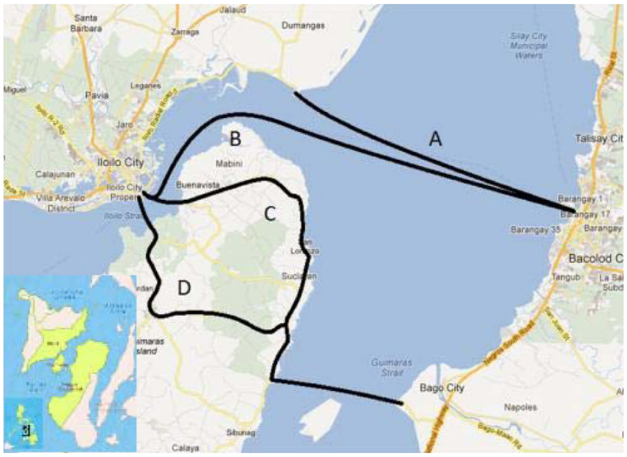


Fig. 1. Major Iloilo-Negros Occidental Travel Routes (Main mode: A - RORO; B - Fastcraft Ferry; C/D - Pumpboat)

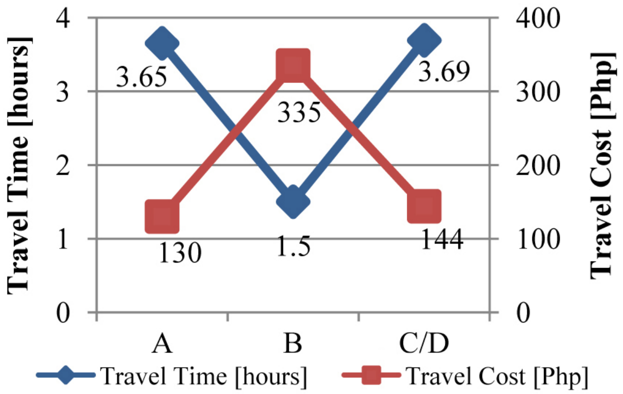


Fig. 2. Characteristics of Iloilo-Negros Occidental Inter-Island Travel Options (Main mode: A - RORO; B - Fastcraft Ferry; C/D - Pumpboat)

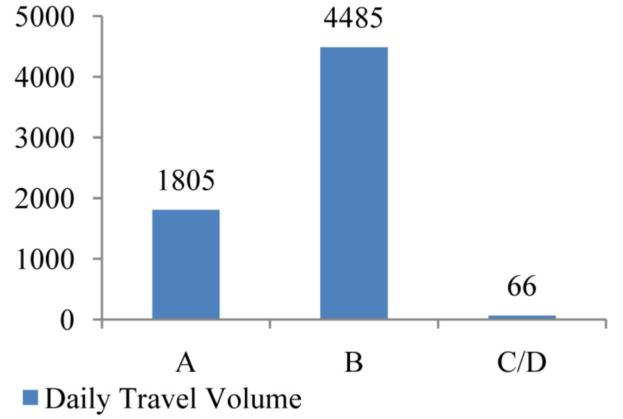


Fig. 3. Daily Travel Volume Using Iloilo-Negros Occidental Travel Options (Main mode: A - RORO; B - Fastcraft Ferry; C/D - Pumpboat)

With an average of 140 trips per week, the Fastcraft ferry (Route B) caters to most of the demand. RORO (roll-on, roll-off) ferry travel, on the other hand, which offers around 100 trips per week on the average, serves as an effective alternative (Route A). This travel can also be made through inter-modal travel through the island of Guimaras. Iloilo–Guimaras passenger travel can be done using pumpboats, embarking from Iloilo City and alighting at either Buenavista (Route C) or Jordan (Route D). Port-to-port transportation across Guimaras island can be made through jeepneys, multicabs, and vans. Guimaras–Negros Occidental travel can then be performed using pumpboats from San Lorenzo to Pulupandan, completing the Iloilo–Negros travel.

The characteristics of the basic travel options for the Iloilo City to Negros Occidental travel are shown in Fig. 2. Fig 3 shows that a great deal of the inter-island travelling population, 70.56%, uses the fastcraft ferry option (Route B). This option has the shortest total travel time and does not involve intermodal transfers. However, this option is the most expensive, costing around more than twice the total travel costs incurred using the nearest alternative. This can mean that the travelling population prioritizes travel time and comfort, in terms of the number of transfers, greatly over travel cost.

III. Model Data

The variables were categorized into a total of 11 categories to simplify the descriptions of the variables, as shown in the Appendix. Also shown, the travel choices were reduced to A, B, and, C, where options C and D were merged into one as almost no data was gathered for the latter.

IV. Logit Modeling

In the development of models, all modeling variables were used in different combinations to come up with the best models possible. In evaluating which models are suitable in describing the travel mode choice of the travelling population, many criteria were considered. First, the coefficients of the variables were checked if the sign (positive or negative) agrees with prior knowledge, considering what quantity the variable is representing (utility or disutility). Furthermore, the coefficients' statistical significance are checked through its respective P-values, log likelihood functions, and Rho-squared measures. Lastly, accuracy of models in predicting the travel choice was considered.

The following multinomial logit (ML) models were developed using NLOGIT, with a logit structure shown in Fig. 4, having only three Alternatives, A, B and C, with Alternative C as the base alternative. Using the logit models, the probability of an individual to choose a particular alternative can be computed using equation (1).

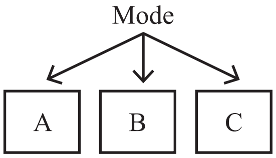


Fig. 4. Multinomial Logit Structure

$$P(j) = \frac{e^{U_j}}{e^{U_A} + e^{U_B} + e^{U_C}} \quad (1)$$

Where: U_j : utility of alternative j
U_A : utility of alternative A
U_B : utility of alternative B
U_C : utility of alternative C

TABLE I.
Multinomial Models Developed with its Variables

Variables	ML1	ML2	ML3
	Coefficient	**Coefficient**	**Coefficient**
A_A	.47441	-1.05824**	3.73793**
A_B	-.87900*	1.76049**	5.23193**
TOTCOST	-.00559**		
TOTTIME	-.48424**		
COMFORT	3.9924**		1.20335**
LNDTIME		-.01342**	
C_TVEH		-.01238**	-.01043**
T_ORPR			-.03467**
WAITTME			.00956**
T_PRDE			-.01168**
AxINC1	.00011**	.00011**	.00011**
AxAGE1	-.05496**	-.05293**	-.05304**
BxINC2	.00013**	.00013**	.00013**
BxAGE2	-.05735**	-.05377**	-.05584**
Goodness of Fit Measures			
L(*B*)	-788.70	-782.86	-634.87
L(0)	-1377.66	-1377.66	-1377.66
-2[L(0)-L(*B*)]	1177.92	1189.60	1479.58
-2[L(C)-L(*B*)]	558.69	570.38	860.35
P^2	0.428	0.432	0.537
$^{-}P^2$	0.262	0.267	0.403

** - passed the 0.1 level of significance*

*** - passed the 0.05 level of significance*

As seen in Table 1, for the ML1 model, TOTTIME, TOTTIME and COMFORT were used as alternative-specific deterministic variables, while LNDTIME and C_TVEH were used in model ML2, and T_ORPR, WAITTME, T_PRDE, C_TVEH and COMFORT for model ML3. For all three models ML1, ML2 and ML3, AGE and INCOME were used as generic deterministic variables. Going over the coefficients, it can be seen that TOTCOST, TOTTIME, LNDTIME, C_TVEH, T_ORPR, and T_PRDE have negative signs, meaning the items are considered disutilities, which follows priori knowledge since these consider values spent by the individual. For the variables COMFORT and WAITTME, the coefficients are positive. As for INCOME and AGE, the coefficients have consistent positive and negative signs, respectively.

Quantities involving cost and time being significant were expected with the common understanding of travel mode choice scenario. These variables involve quantities that are most directly connected to the choice situation as these are directly spent by the individual as a choice decision is made. Comfort being significant with a positive coefficient was also expected. Income was also found to be statistically significant with a positive coefficient. This can be explained simply as the enabling effect of income. People with higher income are less sensitive to higher costs and are capable to pay more, in exchange for other benefits like shorter travel time and/or higher comfort, among others.

Age, in general, was found to be significant, with negative coefficients. This indicates that older people are more likely to use the intermodal option passing through Guimaras province, even though it has significantly higher travel time as compared with the other two alternatives. This can be interpreted to mean that sensitivity to travel time decreases as an individual gets older. This may also be connected to older people being less in a hurry and being less constrained by their schedules. Another possible explanation is the automation of choice decision through practice, where people would prefer using the alternative they had been using long before, for example, in a time where the other two relatively newer options were still unavailable.

V. Neural Network Modeling

In the development of neural network models, the variables were included in sets to simplify the possible combinations of variables. MATLAB was used to generate the neural network models. Table 2 shows the variable categories included in each input data set used, where "1" corresponds to the set being included, and "0," otherwise. As shown, set A has the least number of input variables at a total of only 6 (comprised of 3 travel experience variables and 3 passenger personal information variables), while set P has the most at 74 variables including all the variables available.

As for the target data, three (3) output nodes were set, corresponding to the choice among the three travel choices available (A, B, and C), where a value of "1" corresponds to the passenger's choice, and "0" for the other options that were not chosen. The Levenberg-Marquardt algorithm was used as the training function, as it enables the network to find the solution even if it starts very far off the minimum. The gradient descent method was used for the adaption learning function. The performance function used was the mean square error (MSE). The number of hidden layers was set to only one to come up with just a simple neural network. The input data were disaggregated to training, validation, and testing, with shares of 70%, 15%, and 15%, respectively.

The activation function used in the input-hidden connection was hyperbolic tangent, while sigmoidal logistic was employed in the hidden-output connection. This was done to account for the negative contributions of some variables within the network, but end with a strictly positive output, as the target output is only "1" or "0." In testing to find the best combination of input data and number of hidden neurons, neural network models were developed while varying the number of hidden neurons for each input data set. Fig 5 shows its graphical representation of the R^2 values of the neural networks developed, while Table 3 shows the details.

TABLE II
Variables for Input Data Set

	Variable Category	Trip Purpose	Passenger Travel Information	Travel Experience	Travel Choice Information	Access Information	Egress Information	Others	Passenger Personal Information	Other Passenger Personal Information	Other Passenger Financial Information	General Information	Total Number of Variables
No. of Var		5	5	3	9	9	9	4	3	3	6	18	
Input Data Set	A	0	0	1	0	0	0	0	1	0	0	0	6
	B	0	1	0	0	0	0	0	1	0	0	0	8
	C	0	0	0	1	0	0	0	0	0	0	0	9
	D	0	1	1	0	0	0	0	1	0	0	0	11
	E	0	0	0	1	0	0	0	1	0	0	0	12
	F	1	1	1	0	0	0	0	1	0	0	0	16
	G	0	1	0	1	0	0	0	1	0	0	0	17
	H	1	1	0	1	0	0	0	1	0	0	0	22
	I	1	1	1	1	0	0	0	1	0	0	0	25
	J	1	1	1	0	0	0	0	1	0	0	1	34
	K	1	1	1	1	1	0	0	1	0	0	0	34
	L	1	1	1	1	0	1	0	1	0	0	0	34
	M	1	1	1	1	0	0	0	1	0	0	1	43
	N	1	1	1	1	1	1	0	1	0	0	0	43
	O	1	1	1	1	1	1	1	1	1	0	1	68
	P	1	1	1	1	1	1	1	1	1	1	1	74

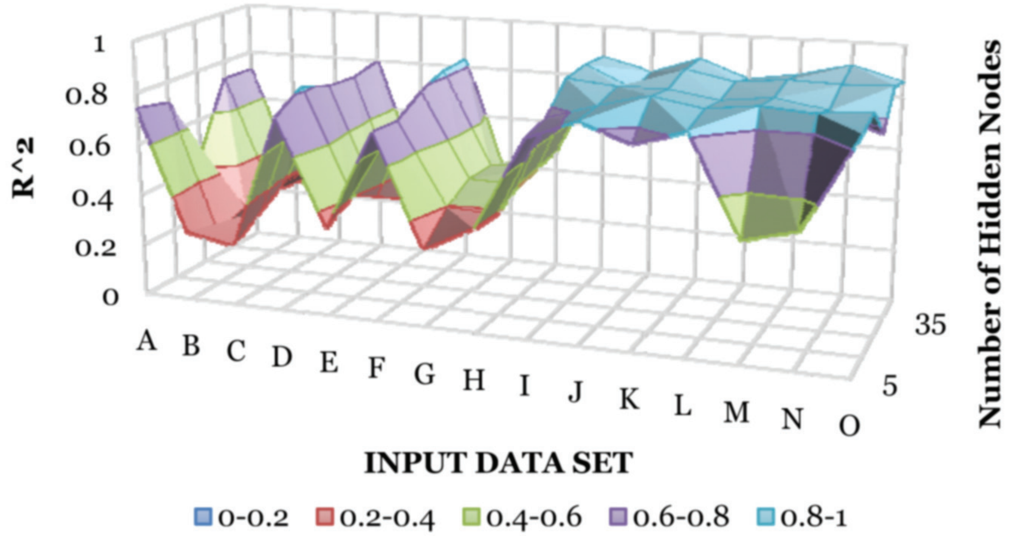


Fig. 5. R^2 Performance of Neural Networks

TABLE III
R^2 PERFORMANCE OF NEURAL NETWORKS

Input Data Set	Number of Hidden Neurons				
	5	15	25	35	45
A	0.738551	0.714211	0.438655	0.725819	0.722636
B	0.267713	0.203022	0.248054	0.261878	0.241258
C	0.234643	0.31351	0.388565	0.383347	0.377549
D	0.744217	0.817523	0.779248	0.770621	0.795272
E	0.342775	0.409037	0.346026	0.279312	0.277824
F	0.724133	0.716291	0.782765	0.822667	0.8281
G	0.296546	0.397316	0.453899	0.454155	0.291665
H	0.395918	0.390238	0.449892	0.464592	0.480915
I	0.747533	0.77171	0.874973	0.886309	0.877257
J	0.823992	0.831197	0.858384	0.803013	0.831598
K	0.758833	0.883487	0.864175	0.89842	0.897453
L	0.822032	0.851634	0.836201	0.861945	0.829247
M	0.453414	0.876358	0.860238	0.883356	0.861407
N	0.503972	0.863896	0.827627	0.877145	0.907847
O	0.856476	0.88298	0.761989	0.887402	0.85855
P	0.846639	0.835762	0.761658	0.8904	0.90117

As shown, sets A, D, F, and I onwards have considerably reliable R^2 values. Going back to Table 2, it can be seen that the similarity of these input data sets is the inclusion of travel experience variables. In set A, where only travel experience and passenger personal information were used, the R^2 values only reached a little over 0.7. If variables on passenger travel information were added as shown in set D, the R^2 values reached 0.8 when the number of hidden neurons was set at 15. When variables dealing with trip purpose were added, the R^2 also attained values over 0.8, but needed more hidden neurons and iterations. Furthermore, when variables are added, the R^2 values tend to show a slight increase, but require significantly longer time for network development. Thus, to have a simple, yet still statistically reliable model, the choices were cut down to sets A, D, and F. Table 4 shows a summary of the variables included in these sets, as well as the R^2 values for the training, validation, and testing of the best neural networks using sets A, D, and F, respectively.

Following the guidelines in the appropriate number of hidden neurons, the three networks were evaluated. The first condition sets the maximum number of hidden neurons to be twice the number of input nodes plus one. The 16-45-3 model does not satisfy this condition (2(16) + 1 = 33 < 45), and is thus, removed. The second guideline states that the number of hidden neurons should be between the average number of input and output nodes and their sum. Both the 6-5-3 and 11-15-3 models satisfy the first part of this condition, but only the 11-15-3 model fails the next (11 + 3 = 14 < 15). However, as the R^2 value of the 6-5-3 model is relatively low, and since the 11-15-3 model only slightly failed to satisfy the guidelines, the latter was chosen as the better model.

To determine the optimum number of hidden neurons, neural network models were developed while varying the number of hidden neurons from 5 to 25. Figure 6 and Figure 7 show the R^2 and mean square error performances of the models, respectively. As shown, the highest R^2 values for training, validation, and testing were attained when the number of hidden neurons was at 15. Also shown, the lowest mean square error was reached with 15 hidden neurons. Thus, this paper recognizes the 11-15-3 neural network (i.e., 11 input variables; 15 hidden neurons; 3 output nodes) as the best model to describe the discrete choice behavior being studied. Figure 8 shows the structure of the best model.

TABLE IV
Best Neural Networks Developed

		6-5-3 NN		**11-15-3**		**16-45-3**	
Variables		Travel Experience	USED_A USED_B USED_C	Travel Experience	USED_A USED_B USED_C	Travel Experience	USED_A USED_B USED_C
		Passenger Personal Information	AGE GENDER INCOME	Passenger Personal Information	AGE GENDER INCOME	Passenger Personal Information	AGE GENDER INCOME
				Passenger Travel Information	NUM_GRP CHL_GRP FREQNCY BEFLNCH WKDAY	Passenger Travel Information	NUM_GRP CHL_GRP FREQNCY BEFLNCH WKDAY
						Trip Purpose	PURWORK PURVACA PURSCHL PURBUSI PURHOME
Input Nodes		6		11		w 16	
Hidden Neurons		5		15		45	
R^2	Training	0.73630I1		0.810594		0.848867	
	Validation	0.751793		0.828082		0.819496	
	Testing	0.735975		0.85705		0.742958	
	All	0.738551		0.820129		0.828100	

The variables used in the final neural network model include passenger travel information (number of people in travel group, number of children in travel group, frequency of travel, time of day, day of week), travel experience information (experience of using options A, B, or C in the past), and passenger personal information (age, gender, income). This does not follow the common idea that travel time and travel cost are the most significant factors contributing to a travel mode choice. As previously mentioned, the statistically acceptable models are those which primarily included travel experience information. This can be interpreted as the neural network's effort to model the behavior and not necessarily the choice decision process.

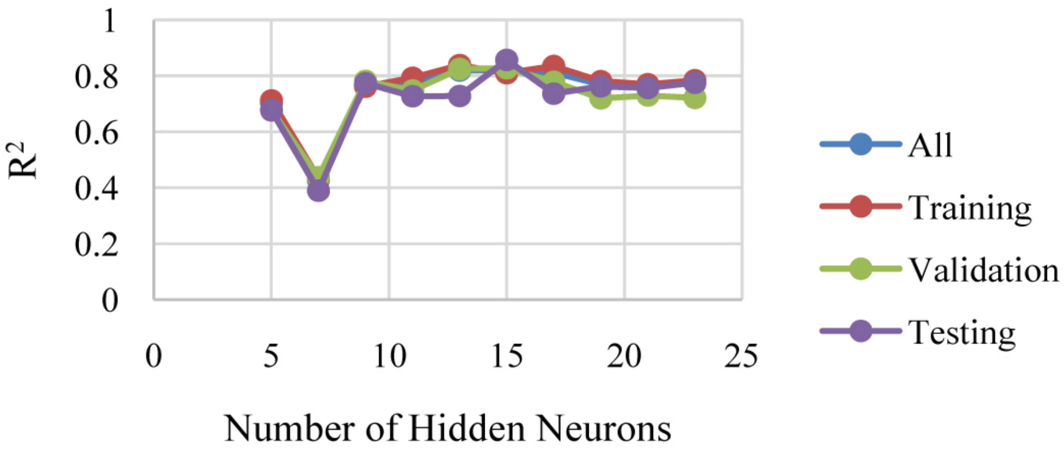


Fig. 6. R^2 Performance of Set D Neural Networks

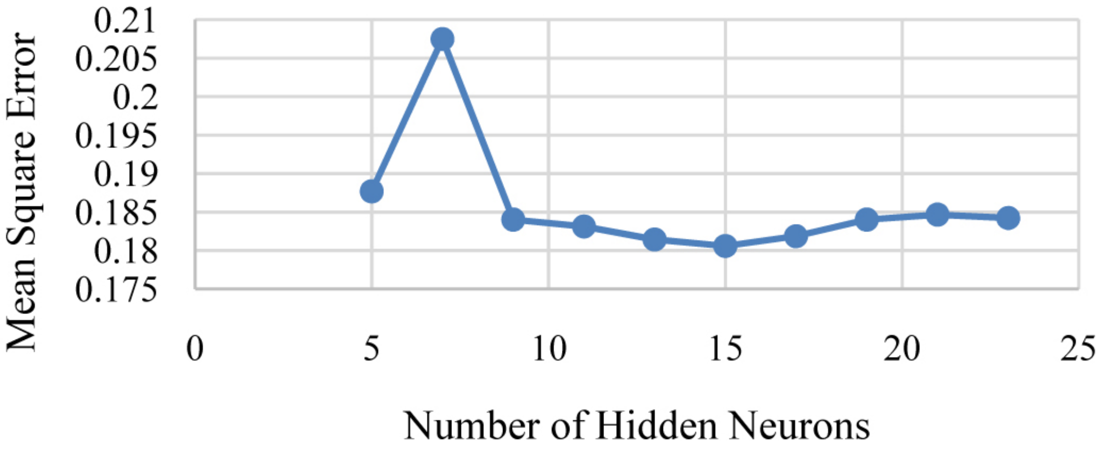


Fig. 7. Mean Square Error of Set D Neural Networks

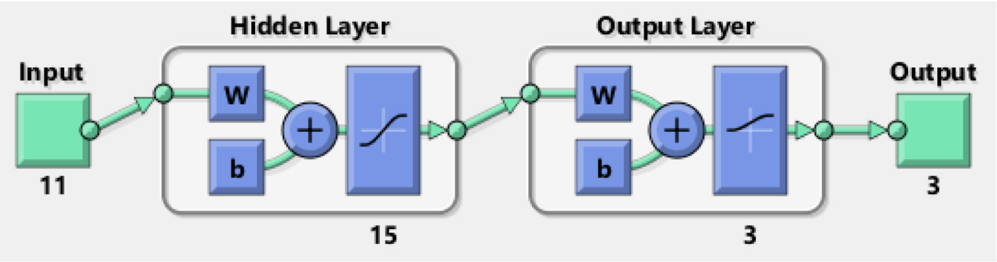


Fig. 8. 11-15-3 Neural Network Structure

VI. Summary

Out of all of the variables found to be significant, only AGE and INCOME were found to be significant in both logit and neural network models. All other variables in the neural network were not found to be significant in the logit models, just as those other variables found to be significant in the logit models were insignificant in the neural network models. This shows that the models developed captured different facets of the same discrete choice situation. The logit models can be understood to be more focused on modeling the decision making process of the passenger, while the neural network concentrated on modeling the overall historical behavior.

TABLE V
Performance of Logit and Neural Network Models

Measure		ML3	11-15-3 NN
Pseudo R^2		0.40277 (From Table 2)	–
R^2		0.80332 (Interpolated)	0.82013 (From Table 5)
Prediction Accuracy [%]		70.33493	92.98246
Choice	A	65.94724	96.16307
	B	77.58621	102.122
	C	26.50602	100.00

Table 5 shows a comparison of the R^2 and prediction accuracy of the best models developed. R^2 values for the logit models were estimated from the established relationship between linear R^2 and logit pseudo-R^2 values, shown in Figure 9. As shown in the table, the best neural network has a higher R^2 value compared with the best ML models. Also, the 11-15-3 NN has the highest prediction accuracy at almost 93%. This shows that the neural network model is a better fit in describing the travel choice behavior of the transport network studied as compared with the multinomial logit model.

Also shown is the disaggregated prediction accuracy of the models, where a 65.95% prediction accuracy means that 65.95% of those who chose option A were predicted to choose option A. As shown, the neural network model is also capable of accurately predicting the minority (Choice C), having a prediction accuracy of 100%, as compared with the 26.51% of both ML3 and NL3. Logit models usually treat the minority as errors. In the neural network, on the other hand, the minority is the one having the perfect prediction rate. This shows that the neural network takes every observation as a true and perfectly valid observation, and thus, tries to model it along with all other observations. The prediction accuracy, computed to be at 102.12% for Choice B, can be explained as the model predicting more individuals choosing option B than the actual number, corresponding to some prediction errors.

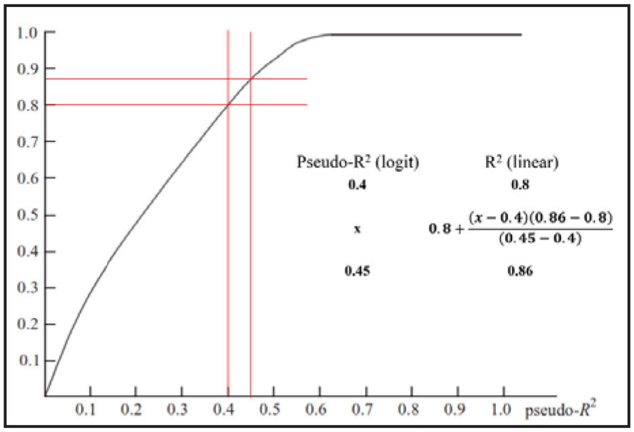


Fig. 9. Relationship of Logit pseudo-R^2 and linear R^2

VII. Conclusions & Recommendations

These findings do not mean that neural networks are always better than logit models. If anything, this paper only shows that neural networks can also be used in modeling intra-regional travel, aside from urban trips that have been the focus of most other researches. Furthermore, the power of logit models to predict travel choices is still valid as it requires less input but yet produces comparable fitness and prediction accuracy.

Also, while the neural network can statistically better model the travel choice being studied, logit models explicitly show the numerical contributions of the variables that ultimately add up to a decision. This allows for the computation of external quantities like the value of time of the population, which can be used in many other applications, unlike the black-box characteristic of neural networks that does not provide any insight on the structure of the function being approximated.

This paper also recognizes the applicability of using data sets in determining the best combinations of input data. As the total number of input variables amount to 74, there would be much difficulty in accounting for all possible combinations. Thus, the researcher found it best to keep the neural network as simple and uncrowded as possible by looking at the small improvements of R^2 values as more input variables and hidden neurons are added. Furthermore, as the research was performed with the aim of finding a more efficient approach in developing discrete choice models, grinding through strenuous modeling using all possible combinations of variables, while finding the optimum number of hidden neurons at the same time, would not have been the way to go.

As for the computation of relative importance of variables, in testing its significance in the discrete choice model, conducting connection weight analysis on the neural network is recommended. As the previous work already has discussions on marginal effects and elasticities for the logit models developed, determining the relative importance of the variables found to be significant in the neural network can be used to further evaluate the applicability of neural networks in predicting travel choices. Being able to get the same findings would only strengthen the idea of the applicability of neural networks in discrete choice analysis.

Acknowledgment

The researchers would like to thank Dr. Andres Winston C. Oreta and Dr. Lessandro Estelito O. Garciano for sharing their knowledge and guidance on the application of neural networks in data modeling.

References

[1] P. Nijkamp, A. Reggiani, and T. Tritapepe. (1996). "Modeling inter-urban transport flows in Italy: A comparison between neural network analysis and logit analysis," *Transportation Research Part C: Emerging Technologies, 4*(6), 323–338. doi:10.1016/s0968-090x(96)00017-4

[2] Y. Bentz, and D. Merunka. (2000). "Neural networks and the multinomial logit for brand choice modeling: A hybrid approach." Retrieved from: cs.uni-muenster.de/Professoren/Lippe/diplomarbeiten/html/eisenbach/UntersuchteArtikel/BeMe00.pdf

[4] G. E. Cantarella, and S. D. Lucas. (2005). "Multilayer feedforward networks for transportation mode choice analysis: An analysis and a comparison with random utility models." *Transportation Research Part C: Emerging Technologies, 13*(2), 121-155. doi:10.1016/j.trc.2005.04.002

[5] P. C. Vythoulkas and Koutsopoulos, H. N. (2003). "Modeling discrete choice behavior using concepts from fuzzy set theory, approximate reasoning and neural networks," *Transportation Research Part C: Emerging Technologies,* 11(1), 51–73. doi:10.1016/s0968-090x(02)00021-9

[6] A. Norets. (2008). "Estimation of dynamic discrete choice models using artificial neural network approximations." Retrieved from: www.econ.brown.edu/fac/Andriy_Norets/papers/anorets07_ANNMCMC.pdf

[7] S. Nakayama, J. Takayama, and Y. Yamashita. (2008). "A non-linear analysis of discrete choice behavior by the logit model." Retrieved from: www.trip.t.u-tokyo-ac.jp/eastsjwp/2008/0804/0804.pdf

[8] H. Dia. (2010). "Evaluation of discrete choice and neural network approaches for modeling driver compliance with traffic information." Retrieved from: www.researcgate.net/publlication/51028304_Evaluation_of_discrete_choice_and_neural_network_approaches_for_modeling_driver_compliance_with_traffic_information

[9] S. Pulugurta, A. Arun and M. Errampalli. (2013). "Use of artificial intelligence for mode choice analysis and comparison with traditional multinomial logit model." *Procedia – Social and Behavioral Sciences,* 104*,* 583–592.

[10] Roquel, K., & Fillone, A. (2013). "Mode choice analysis of inter-island passenger travel from Iloilo to Negros Occidental, Philippines." Retrieved from: easts.info/on-line/proceedings/vol9/PDF/P125.pdf

Appendix

Category	Variable	Description
Trip Purpose	purwork	1 – trip purpose is work; 0 – if not
	purvaca	1 – trip purpose is vacation; 0 – if not
	purschl	1 – trip purpose is school; 0 – if not
	purbusi	1 – trip purpose is business; 0 – if not
	purhome	1 – trip purpose is home; 0 – if not
Passenger Travel Information	num_grp	Number of people in travel group
	chl_grp	Number of children in travel group
	freqncy	Frequency of travel
	beflunch	1 – time of travel is before 12:00 P.M.; 0 – if not
	wkday	1 – day of travel is a weekday; 0 – if not
Travel Experience	usedrta	1 – have experience using route A; 0 – if none
	usedrtb	1 – have experience using route B; 0 – if none
	usedrtc	1 – have experience using route C; 0 – if none
Travel Choice Information	a_time	Travel time when using option A
	b_time	Travel time when using option B
	c_time	Travel cost when using option C
	a_tcost	Travel cost when using option A
	b_tcost	Travel cost when using option B
	c_tcost	Travel time when using option C
	a_wttme	Waiting time when using option A
	b_wttme	Waiting time when using option B
	c_wttme	Waiting time when using option C
Access Information	a_comorpr	Comfort of accessing option A
	b_comorpr	Comfort of accessing option B
	c_comorpr	Comfort of accessing option C
	a_torpr	Time of accessing option A
	b_torpr	Time of accessing option B
	c_torpr	Time of accessing option C
	a_corpr	Cost of accessing option A
	b_corpr	Cost of accessing option B
	c_corpr	Cost of accessing option C
Egress Information	a_comprde	Comfort of egressing option A
	b_comprde	Comfort of egressing option B
	c_comprde	Comfort of egressing option C
	a_tprde	Time of egressing option A
	b_tprde	Time of egressing option B
	c_tprde	Time of egressing option C
	a_cprde	Cost of egressing option A
	b_cprde	Cost of egressing option B
	c_cprde	Cost of egressing option C

Others	a_cbag	Additional cost for baggage when using option A
	b_cbag	Additional cost for baggage when using option B
	c_cbag	Additional cost for baggage when using option C
	b_rdtrp	1 – option B user bought roundtrip tickets; 0 – if not
Passenger Personal Information	age	Age of passenger
	gender	1 – passenger is male; 0 – if female
	income	Personal monthly income of passenger
Other Passenger Personal Information	single	1 – passenger is single; 0 – if not
	married	1 – passenger is married; 0 – if not
	num_chl	Number of children of passenger
Other Passenger Financial Information	num_mot	Number of motorcycles owned by passenger
	num_car	Number of cars owned by passenger
	num_van	Number of vans owned by passenger
	num_suv	Number of SUVs owned by passenger
	num_jpn	Number of jeepneys owned by passenger
	vacatn	Number of vacations passenger takes yearly
General Travel Choice Information	a_totcom	Total comfort when using option A
	b_totcom	Total comfort when using option B
	c_totcom	Total comfort when using option C
	a_lndtime	Total time travelling on land when using option A
	b_lndtime	Total time travelling on land when using option B
	c_lndtime	Total time travelling on land when using option C
	a_seatime	Total time travelling at sea when using option A
	b_seatime	Total time travelling at sea when using option B
	c_seatime	Total time travelling at sea when using option C
	a_freqncy	Operation frequency of option A
	b_freqncy	Operation frequency of option B
	c_freqncy	Operation frequency of option C
	a_tottime	Total time when using option A
	b_tottime	Total time when using option B
	c_tottime	Total time when using option C
	a_aircon	Time spent in air-conditioned facility when using option A
	b_aircon	Time spent in air-conditioned facility when using option B
	c_aircon	Time spent in air-conditioned facility when using option C

Selection of Artificial Neural Network Training Algorithms in the Detection and Classification of Wavelet de-noised Musical Tone Stimulated EEG Signals

Roy Francis Navea and Elmer Dadios

Abstract—**The human brain can be stimulated by internal and external factors with which the effect of these can be traced from brainwaves or EEG signals. The natural complexity of EEG signals calls for methods by which information can be extracted and used for a particular purpose. In this study, musical tones were used to stimulate the brain and an attempt was made to detect and classify these stimulations from the EEG signals. An Artificial Neural Network (ANN)-based classifier was employed to do this task. Wavelet based de-noising was used to smoothen the musical tone stimulated EEG signals and among the 110 known mother wavelets, the reverse biorthogonal 'rbio3.1' and 'rbio3.3' using the 'rigrsure' thresholding method satisfied the selection criteria for better de-noising effects.**

Detection and classification were performed using ANNs implementing four different training algorithms. Results show that *trainbr* or *trainlm* is good for detection while the *trainlm* was found to be better than the other training algorithms used when it comes to classification. The metrics for selecting the training algorithm were based on the F-score and the rejection rate having the condition that F-score should be high while the rejection rate should be low.

Keywords—**Electroencephalogram, musical tone stimulation, wavelet de-noising, training algorithms, Artificial Neural Networks**

Roy Francis Navea, Electronics and Communications Engineering Department De La Salle University Manila, Philippines (e-mail: roy.navea@dlsu.edu.ph)

Elmer Dadios, Manufacturing Engineering and Management Department, De La Salle University, Manila, Philippines (e-mail: elmer.dadios@dlsu.edu.ph)

I. Introduction

THE relaxed state of the mind can be disturbed by different stimulation causing a response that can be mapped and viewed through electroencephalography (EEG). The stimulation can be assessed and characterized [1] using different signal processing techniques. Not just stimulations, inherent motions [2],[3] and other regular activities of the human body [4]–[6] are deeply registered and appear in different patterns in the brain. Different attempts were made to understand brainwave patterns according to a specific task or stimulation [6]–[8] and this brought about a variety of approaches that addresses the nature of the task and stimulation.

In this study, an attempt was made to detect and classify the disturbance caused by musical tone stimulation by utilizing different algorithms in an artificial neural network. Musical tones are the building components of a melody when they are arranged in a specific timing pattern guided by a time signature. Lyrics were added to give meaning to these arrangement, thus, producing a song. The uniqueness of this study is that it focuses on the music itself, specifically the pitch, and not on the song which is a short piece of music with lyrics that comes with different genres. Stimulated EEG signals were used to investigate the relative effect of the musical tones [1] through the different features which can be extracted from it. EEG signals in its raw form requires a number of processing techniques in order reduce its complexity and utilize it into something useful and informative.

Processing EEG signals poses challenges in developing algorithms by which they can be utilized for a specific purpose. EEG signal patterns can be used as a basis for diagnoses [5], [7], [9]–[12] and a control signal for actuators and motors [13], [14]. Before doing so, preprocessing has to take place to remove unwanted details in the EEG stream. Hence, filtering techniques have to be performed.

One of the best de-noising methods is wavelet-based filtering due to its capability to deal with both time and frequency maps of the given signal simultaneously as compared to Fourier-based filters which suffer from substantial loss of EEG data [15]. It is an important matter how mother wavelet (MW) and thresholding method is selected.

Detection and classification are always paired with feature extraction. Features are the characteristics of the signal of interest which discriminate it from the others. Features can be extracted using different algorithms which includes both time-domain, frequency-domain and statistical characteristics [4], [16]–[19]. For as long as these features and characteristics can possibly differentiate one segment from another, they are good inputs to the classifier engine. Power and energy features [1], [2],[20] are useful features since stimulation is basically a transfer of energy from the stimuli to the receptors (or human sensory organs) which generate impulses that travel through the nerves to the brain.

Features are fed into classifiers that come in a variety of types and algorithms used. Some of the well-known classifiers include Artificial Neural Networks (ANN), Naïve Bayes (NB) classifiers, k-nearest neighbor (k-NN) classifiers and Support Vector Machines (SVM). In [21], EEG signals from epileptic patients were used. The features are based on Discrete Fourier Transform (DFT) coefficients and results show that NB classifiers is better than k-NN when it comes to classification accuracy and computation time. In [22], single EEG channel was used to classify levels of drowsiness. Features used are based on Fast Fourier Transform (FFT) coefficients and results show that ANN is better than SVM in terms of accuracy and receiver operating characteristic (ROC) curve. Each classifier may perform better than the other depending on the application and type of signal used [21]–[23].

This study focuses on the implementation gradient descent, quasi newton, conjugate gradient and Bayesian regularization ANN algorithms using the training functions (TF), *trainrp*, *trainlm*, *trainscg*, and *trainbr*.

I. Methods

A. Audio Stimulus and Data Gathering Procedures

The audio stimulus is composed musical tones in the key of C. The tones are C, F, and G are located at the 4th octave of a standard piano keyboard. The tones are arranged in a musical piece [1],[20] as shown in Figure 1. Rests (whole, half and quarter rests) are periods of silence while the notes (half notes) are the tones. The long series of rests before the first tone establishes the baseline (baseline1) while the rests that come immediately after a note is the secondary baseline (s-baseline).

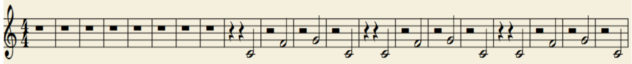


Fig. 1. Audio Stimulus Piece

A timing table [1],[20], as shown in Table I, was used to easily determine where in time a tone was played and stimulated the brain. No delays were assumed. The timing table is the summary of the audio stimulus in terms of the stimuli, time stamp, period, number of samples and sample series. The stimuli were named baseline1, s-baseline, C, F, and G. The audio was played for 3 minutes and 48 seconds. Baseline1 has the longest period with 180 seconds. S-baseline and the notes have a period of 2 seconds for each occurrence. The EEG signal was sampled at 128 samples per second. Baseline 1 has the largest number of samples with 23040. S-baseline and the notes have 256 samples each. There are 29184 samples corresponding to the total period of the audio stimulus. Each stimulus was mapped in the sample series for segmentation purposes.

The data used were taken from 15 undergraduate students with ages typically ranging from 18 to 21. As in [20], the data gathering was performed in a dim-lighted acoustically prepared room. The respondents were seated one at a time and were asked to close their eyes to minimize eye-related artefacts. An ear phone was used for optimal audio reception. A 14-channel Emotiv EPOC neuroheadset was used and its signal quality and data transmission functionality was carefully monitored through its graphical user interface.

B. Detection and Classification Procedures

The general detection and classification process follows the flowchart in Figure 2. The raw EEG signals obtained from the neuroheadset were loaded in Matlab®.

The signals were bandpass filtered within the alpha (8Hz–13Hz) and beta (13Hz–30Hz) bands, and smoothened using wavelet de-noising techniques. Two classifiers were used. The first one (ANN1) was for detecting the tone-stimulated EEG signal and the second one (ANN2) is to classify it according to C, F or G tone. The display process is an indicator of what has been detected or classified. There is a possibility that a certain signal might not be classified [24] and to address this, rejection ratios were indicated in the results.

TABLE I

Audio Stimulus Timing Table

Stimuli	baseline 1	s-baselline	C	s-baseline	F	s-baseline	G
Time Stamp	0-3:00	3:01-3:02	3:03-3:04	3:05-3:06	3:07-3:08	3:09-3:10	3:11-3:12
Period	180 sec	2 sec	2 sec	2 sec	2 sec	2 sec	2 sec
No. of Samples	23040	256	256	256	256	256	256
Sample Series	1-23040	23041-23296	23297-23552	23553-23808	23809-24064	24065-24320	23421-24576

Stimuli	s-baseline 1	C	s-baseline	C	s-baseline	F	s-baseline
Time Stamp	3:13-3:14	3:15-3:16	3:17-3:18	3:19-3:20	3:21-3:22	3:23-3:24	3:25-3:26
Period	2 sec	2 sec	2 sec	2 sec	2 sec	2 sec	2 sec
No. of Samples	256	256	256	256	256	256	256
Sample Series	24577-24832	24833-25088	25089-25344	25345-25600	25601-25856	25857-26112	26113-26368

Stimuli	G	s-baseline	C	s-baseline	C	s-baseline	F
Time Stamp	3:27-3:28	3:29-3:30	3:31-3:32	3:33-3:34	3:35-3:36	3:37-3:38	3:39-3:40
Period	2 sec	2 sec	2 sec	2 sec	2 sec	2 sec	2 sec
No. of Samples	23040	256	256	256	256	256	256
Sample Series	26369-26624	26625-26880	26881-27136	27137-27392	27393-27648	27649-27904	27905-28160

Stimuli	s-baseline	G	s-baseline	C
Time Stamp	3:41-3:42	3:43-3:44	3:45-3:46	3:47-3:48
Period	2 sec	2 sec	2 sec	2 sec
No. of Samples	256	256	256	256
Sample Series	28161-28416	28417-28672	28673-28928	28929-29184

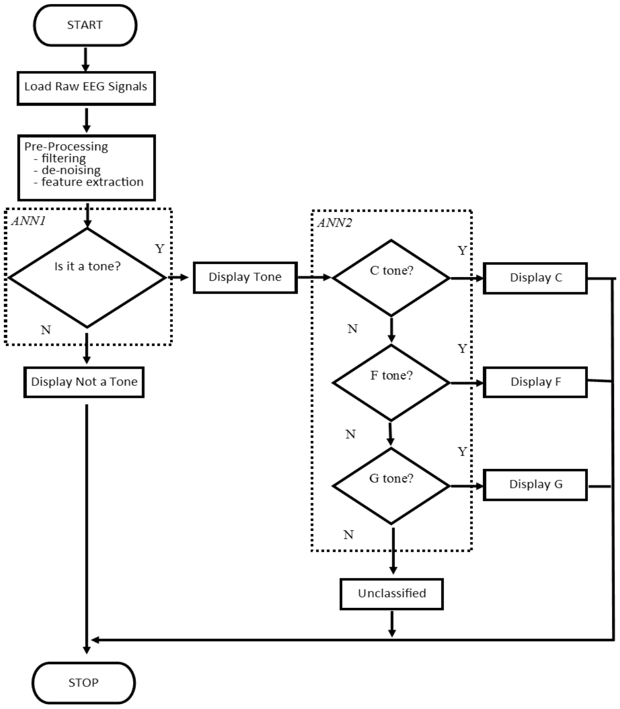


Fig. 2. Detection and Classification General Flowchart

C. *Wavelet-based De-noising*

This filtering technique is a three-step process that includes signal decomposition using DWT by selecting a mother wavelet and the number of decomposition levels, perform thresholding in the wavelet domain and shrink the coefficients by thresholding, and then reconstruct the signal from the thresholded DWT coefficients.

TABLE II

Wavelet Families

Wavelet Family	Wavelet Tag	Count
Daubechies	db1 - db45	45
Coiflet	coif1 - coif5	5
Biorthogonal	bior1.1 - bio6.8	15
Reverse-Biorthorthogonal	rbio1.1 - rbio6.8	15
Discrete Meyer	dmey	1
Symlets	sym2 - sym30	29
Total		**110**

There are 110 known mother wavelets, as shown in Table II, and these were all tested using a soft thresholding algorithm implementing four thresholding methods namely: “rigrsure,” “heursure,” sqtwolog,” and “minimaxi” [25],[26]. A 2-level decomposition was implemented since the baseband signal ranges from 8 Hz–30 Hz covering the alpha and the beta bands. This results to alpha waves ranging from 8Hz–13.5 Hz and beta waves ranging from 13.5 Hz to 19 Hz and 19 Hz to 30 Hz.

Mother wavelet and thresholding method selection was based on the following: signal-to-noise ratio (SNR), peak signal-to-noise ratio (PSNR), mean square error (MSE) and the correlation coefficient (R). These parameters were calculated using (1)–(4), respectively. The original EEG signal is *x(n)* while the de-noised EEG signal is $x_d(n)$. As a selection requirement, SNR, PSNR and R should be at maximum while MSE should be at minimum [25].

$$\text{SNR} = 10\log\left[\frac{\sum_{i=1}^{N}\left(x_d\left(n\right)\right)^2}{\sum_{i=1}^{N}\left(x\left(n\right)-x_d\left(n\right)\right)^2}\right] \tag{1}$$

$$\text{PSNR} = 20\log\left[\frac{\max\left[x\left(n\right)\right]}{RMSE}\right] \tag{2}$$

$$\text{MSE} = \left[\frac{\sum_{i=1}^{N}\left(x\left(n\right)-x_d\left(n\right)\right)^2}{N}\right] \tag{3}$$

$$\text{R} = \left[\frac{cov\left(x\left(n\right)-x_d\left(n\right)\right)}{\sqrt{var\left(x\left(n\right)\right)}\,var\left(x_d\left(n\right)\right)}\right] \tag{4}$$

D. *Feature Extraction*

Feature extraction is the process of determining a unique characteristic, a special feature or a distinct feature vector from a pattern vector. Features are usually divided into the statistical characteristics and the syntactic descriptions. Not all features are good discriminants and all features need not to be used for classification. Large feature vectors require more processing time and are computationally expensive. Best features can be identified according to various criteria and optimization techniques [27]. In classification tasks like using ANN, feature extraction and selection plays an important role.

In this study, four features were considered. The features were based on the statistical characteristics of the power spectrum vectors of the EEG signal and the signals’ energy obtained from autocorrelation.

- Kurtosis: This is a statistical measure of the flatness or peaks of a signal distribution. The kurtosis of the power spectrum vector of the EEG signal was used as a feature [1].
- Skewness: This is another statistical measure that deals with the asymmetry of a signal distribution. As with kurtosis, the skewness of the power spectrum vector of the EEG signal was used [1].
- Power Spectrum Vectors: These are derived from the Hamming-windowed Fourier transform coefficients of the EEG signal. The power spectrum vectors were decomposed using Singular Value Decomposition (SVD) to represent a single feature [20].
- Signal Energy: This is obtained by taking the element at the origin of the autocorrelation sequence of the signal. The autocorrelation of the signal x(n) is defined by

$$r_{xx}\left(l\right) = \sum_{n=-\infty}^{\infty} x\left(n+l\right)x\left(n\right),\ \ l = 0, \pm 1, +2, \ldots \tag{5}$$

where the signal energy is $r_{xx}(0)$.

E. *Artificial Neural Networks (ANN)*

An artificial neural network is composed of ‘neurons’ or ‘cells’ which are linked together by weighted connections. [28]. These units receive input signals from other units or sources and use it to compute for an output signal which is transmitted to other units.

ANNs have three useful layers: the input layer (which receives data from external sources), the hidden layer (which contains internal network input and output data) and the output layer (which sends the output/resulting data). A three-layer feed-forward neural network was used in this study [19]. Feed forward networks are straight forward networks in which data flow from the input side to the output side of the ANN.

The architecture of the network is shown in Figure 3. There are two networks used, one for detection and one for classification. For detection, the input layer is composed of 4 neurons which correspond to the 4 features of the EEG signals. Each input neuron is linked to the 50 sigmoid neurons that forms the hidden layer. The hidden neurons are linked to the output layer which is composed of 2 linear output neurons that correspond to Not Tone (Baseline) or Tone. Same hidden layers were used for classification except that the input layer has 2 neurons corresponding to the energy and power of the signal, and the output layer with three neurons, corresponding to the tones C, F, and G.

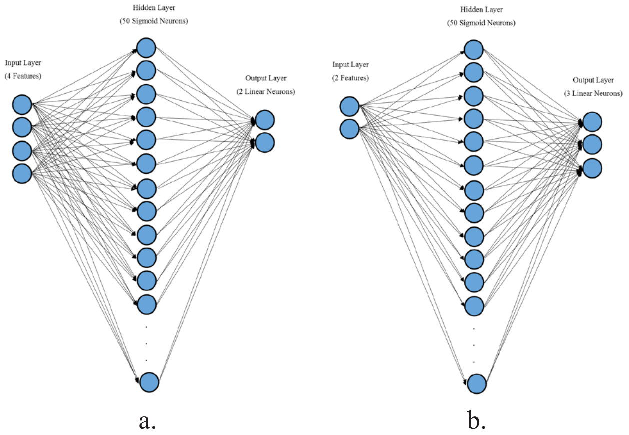


Fig. 3. Neural Network Architecture, a) detection; b) classification

The network was trained using gradient descent, quasi-newton, conjugate gradient and Bayesian regularization by implementing the training functions *trainrp*, *trainlm*, *trainscg* and *trainbr*, respectively [29]. These training functions has the property to deal with the non-linear nature of the EEG signals using non-linear activation or kernel functions. In resilient backpropagation (trainrp), the weight and bias values were updated by using the sign of the partial derivatives leaving its magnitude of no significant effect. This performs faster than a standard steepest descent algorithm. For scaled conjugate gradient Bayesian regulation backpropagation, the weight and bias values are updated according to the Levenberg-Marquardt (LM) optimization method [30] which minimizes squared errors and weights combinations [31]. In Bayesian regulation, weights are introduced into the training objective function denoted by

$$F(w) = \alpha S_w + \beta S_D \tag{5}$$

where S_w is the sum of the squared network weights and S_D is the sum of the network errors. The objective function parameters are defined by the variables α and β. The weights of the network are randomly selected and a Gaussian distribution of the network weights and training set is assumed.

The objective function parameters, α and β, are defined using the Baye's theorem which basically shows the relationship between two variables, say A and B, according to their prior and posterior probabilities [32]. The posterior probability of A with respect to B is defined by

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{6}$$

where P(B | A) is the prior of B conditional to A, P(A) and P(B) are the prior probabilities of A and B not equal to zero, respectively. The optimal weight space can be obtained by minimizing the objective function in (5) which means maximizing the posterior probability function which is given by

$$P(\alpha,\beta|D,M) = \frac{P(D|\alpha,\beta,M)P(\alpha,\beta|M)}{P(D|M)} \tag{7}$$

where α and β are the variables to be optimized, D refers to the weight distribution, M is the specific neural network architecture, P(D|M) is the factor of normalization, P(α, β|M) is the regularization parameters' constant prior density and P(D|α, β, M) is the similarity function of D given α, β, and M. This process results to optimum values of α and β for a given weight space. The LM phase calculates the squared second-order partial derivatives of the objective function (the Hessian) and minimizes the objective function by updating the weights. For non-convergence, the algorithm makes an estimation for new values of α and β. This process repeats itself until convergence is reached [33].

F. Confusion Matrices

Confusion matrices are used to assess the performance of different classifiers [34],[35]. These matrices provide information that leads to determining the sensitivity, specificity, precision, accuracy and F-score of the classifier. Precision tells how many of the positively classified were relevant, sensitivity / recall tells how good a test is in detecting the positives, and specificity is an indicator of how good a test is in avoiding false detections. The harmonic mean of precision and sensitivity is known as the F-score. This is commonly used as a discriminating factor to describe a good classifier. There were instances in which not all of the samples were classified. Hence, the rejection rate of the classifier has to be considered [24].

III. Results and Discussion

As a requirement, higher SNR, PSNR and correlation coefficient, and lower MSE indicates better de-noising effect. Table III shows the maximum values for the SNR, PSNR and R, the minimum values for the MSE, and the mother wavelet where they were obtained with respect to the five segments. Among the four thresholding methods, the "rigrsure" outperformed the other methods. Results show that the mother wavelet that mostly satisfied the conditions were 'rbio3.1' and 'rbio3.3.'

The two identified mother wavelets were then used to de-noise the EEG signals using the 'rigrsure' thresholding method. A sample of an original signal and de-noised signal is shown in Figure 4.

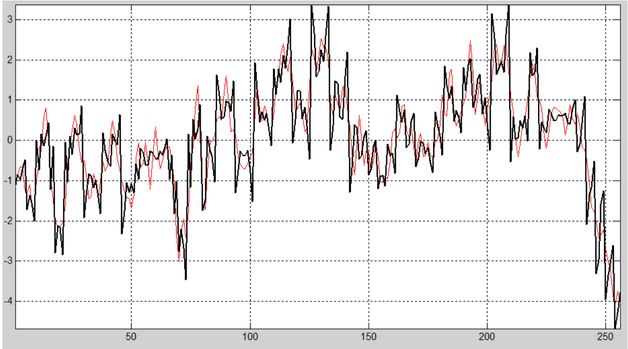


Fig. 4. Original (light) and de-noised (dark) signal

Four features were used to detect whether the EEG is tone stimulated or not, and two features (power and energy) were used to classify the tones whether C, F, or G. Two ANNs (ANN1 and ANN2) were used to perform detection and classification, respectively, implementing the training functions *trainrp*, *trainlm*, *trainscg* and *trainbr* one at a time. Ten training repetitions were performed and the trained network with the highest F-score and lowest rejection rate was selected. Results are shown in Table IV and Table V for detection and classification, respectively.

For detection, the *trainbr* function has the lowest rejection rate and has 0.8571 (85.71%) F-score for both mother wavelets. However, this training algorithm took more time (more or less 120 sec) to converge during simulation.

TABLE III

Segment Parameter Values and Significant wavelets

Thresholding	Parameters	Segments									
		BL		sBL		C		F		G	
Heursure	SNR(dB)	22.0950	rbio3.3	23.0178	rbio3.1	25.2446	rbio3.1	23.5344	rbio3.1	25.8607	rbio3.1
	PSNR (dB)	51.4410	bior3.9	47.4349	rbio3.1	48.8130	rbio3.1	49.1932	rbio3.1	51.7892	rbio3.1
	MSE	0.2455	rbio3.3	0.3368	rbio3.1	0.2729	rbio3.9	0.3043	rbio3.1	0.2421	rbio3.1
	R	0.9956	rbio3.3	0.9945	rbio3.1	0.9959	rbio3.1	0.9934	rbio3.1	0.9966	rbio3.1
Minimaxi	SNR(dB)	20.6370	db39	18.8722	db43	19.4585	db1	17.6053	db1	18.8816	db1
	PSNR (dB)	50.6617	db37	45.6015	db43	42.8141	db31	44.1963	db20	45.7596	db43
	MSE	0.3091	db39	0.4860	db43	0.4481	db1	0.5558	db23	0.5942	db1
	R	0.9944	db39	0.9926	db43	0.9886	coif4	0.9884	db38	0.9903	db41
Rigsure	SNR(dB)	22.3970	rbio3.3	23.9690	rbio3.1	25.8157	rbio3.1	24.5010	rbio3.1	26.1473	rbio3.1
	PSNR (dB)	51.5537	bior3.9	48.6092	rbio3.7	49.1534	rbio3.1	50.4841	rbio3.3	53.4510	rbio3.1
	MSE	0.2251	rbio3.3	0.2299	rbio3.9	0.1756	rbio3.1	0.1750	rbio3.3	0.1459	rbio3.3
	R	0.9958	rbio3.3	0.9961	rbio3.1	0.9957	rbio3.1	0.9963	rbio3.1	0.9978	rbio3.3
sqtwolog	SNR(dB)	15.6564	bior3.9	14.2099	bior3.7	12.7085	bior3.7	12.4987	bior3.7	12.6978	bior3.9
	PSNR (dB)	45.0140	bior3.9	40.2942	bior3.9	36.7208	bior3.7	38.6902	bior3.7	38.5500	bior3.9
	MSE	1.2286	bior3.9	1.5163	bior3.7	1.7650	bior3.7	1.8006	bior3.9	1.9498	bior3.9
	R	0.9753	bior3.9	0.9755	bior3.7	0.9606	bior3.7	0.9669	bior3.7	0.9690	bior3.9

TABLE IV

Precision, Sensitivity, Specificity, Accuracy, F-score, and Rejection rate Table for Detection

MW	TF	Precision	Sensitivity	Specificity	Accuracy	F-score	Rej. Rate
rbio3.1	train1m	76.27%	100.00%	51.72%	81.08%	86.54%	1.33%
	trainscg	75.00%	100.00%	50.00%	80.00%	85.71%	0.00%
	trainbr	75.00%	100.00%	50.00%	80.00%	85.71%	0.00%
	tainrp	76.79%	100.00%	53.57%	81.69%	86.87%	5.33%
rbio3.3	train1m	79.63%	95.56%	63.33%	82.67%	85.86%	0.00%
	trainscg	75.86%	100.00%	51.72%	80.82%	86.27%	2.67%
	trainbr	75.00%	100.00%	50.00%	80.00%	85.71%	0.00%
	tainrp	76.27%	100.00%	48.15%	80.56%	86.54%	4.00%

An alternative training function, *trainlm*, can be considered because this is faster and has an F-score and rejection ratio not significantly far from *trainbr*. The mother wavelet 'rbio3.3' is a better choice because of minimal rejection rate.

For classification, the *trainlm* function has the lowest rejection rate and has highest F-score among all the other training algorithms for both mother wavelets. Hence, 'rbio3.3' is a better choice since it has higher F-score and lower rejection ratio when classified using the *trainlm* function.

The detection and classification of the disturbance caused by the musical tones was successfully performed using the ANN *trainlm* function with de-noised EEG signals using the 'rbio 3.3' mother wavelet. The response of the brain as shown in the EEG signals were characterized in terms of the features extracted from them. These features describing the response were found to be useful enough to differentiate each stimulation whether there is a tone or none and if there is a tone stimulation, whether C, F, or G.

TABLE V
Precision, Sensitivity, Specificity, Accuracy, F-score, and Rejection rate Table for Classification

MW	TF	Segments	Precision	Sensitivity	Specificity	Accuracy	F-score	Rej. Rate
rbio3.1	train1m	C	100.00%	60.00%	100.00%	85.19%	75.00%	9.71%
		F	55.56%	76.92%	61.90%	67.65%	64.52%	
		G	63.64%	58.33%	80.00%	71.88%	60.87%	
	trainscg	C	60.00%	30.00%	81.82%	57.14%	40.00%	9.71%
		F	30.00%	75.00%	30.00%	42.86%	42.86%	
		G	75.00%	27.27%	90.00%	57.14%	40.00%	
	trainbr	C	100.00%	13.33%	100.00%	60.61%	23.29%	0.00%
		F	38.46%	100.00%	17.24%	45.45%	55.56%	
		G	75.00%	20.00%	94.44%	60.61%	31.58%	
	trainrp	C	100.00%	36.36%	100.00%	73.08%	53.33%	12.37%
		F	48.15%	100.00%	30.00%	57.58%	65.00%	
		G	100.00%	22.22%	100.00%	73.08%	36.36%	
rbio3.3	train1m	C	83.33%	76.92%	91.67%	86.49%	80.00%	5.17%
		F	73.33%	91.67%	84.00%	86.49%	81.48%	
		G	96.67%	78.57%	95.45%	88.89%	84.62%	
	trainscg	C	87.50%	58.33%	94.12%	79.13%	70.00%	8.65%
		F	50.00%	91.67%	52.17%	65.71%	64.71%	
		G	83.33%	41.67%	94.74%	74.19%	55.56%	
	trainbr	C	*	0.00%	100.00%	50.00%	0.00%	0.00%
		F	33.33%	100.00%	0.00%	33.33%	50.00%	
		G	*	0.00%	100.00%	50.00%	0.00%	
	trainrp	C	100.00%	45.45%	100.00%	76.00%	62.50%	15.96%
		F	54.55%	92.31%	41.18%	63.33%	68.57%	
		G	66.67%	33.33%	94.44%	79.17%	44.44%	

The confusion matrices for detection and classification using the training algorithms *trainbr* and *trainlm* are shown in Table VI and Table VII. The inputs here served as the bases for the computation of the parameters in the previous tables. One noticeable information is the total summation of the horizontal data. Not tone (NT) has a total of 30 hits while tone (T) has 45 hits. The tones C, F and G has a total of 15 hits each. In the event that the sum of the horizontal hits are less than as mentioned, then, the difference was accounted and was used to determine the rejection rate of the classifier.

IV. Conclusion and Future Directives

Wavelet based de-noising was implemented to smoothen the musical tone stimulated EEG signals. It was found out that the mother wavelets 'rbio3.1' and 'rbio3.3' using the 'rigrsure' thresholding method satisfied the selection criteria in order to provide a better de-noising effect.

Detection and classification were performed using ANNs implementing four different training algorithms. Results show that *trainbr* is good for detection but converges slower. Hence, the *trainlm* is recommended to be an alternative training algorithm. For classification, the *trainlm* was found to be better than the other training algorithms used. The metrics used for selecting the training algorithm were the F-score and the rejection rate which accounts the missed hits of the classifier. F-score should be high while the rejection rate should be low.

Future works may consider other training algorithms for ANN and other classifiers such as SVM, NB and k-NN for detection and classification of musical tone stimulated EEG signals.

TABLE VI
Confusion Matrices for Detection

		Target Class	
		NT	**T**
Output Class	NT	15	15
	T	0	45

a. Detection using rbio 3.1 (trainbr)

		Target Class	
		NT	**T**
Output Class	NT	15	14
	T	0	45

c. Detection using rbio 3.1 (train1m)

		Target Class	
		NT	**T**
Output Class	NT	15	15
	T	0	45

b. Detection using rbio 3.3 (trainbr)

		Target Class	
		NT	**T**
Output Class	NT	13	16
	T	0	45

d. Detection using rbio 3.3 (train1m)

TABLE VII
Confusion Matrices for Classification

		Target Class		
		C	**F**	**G**
Output Class	C	10	3	0
	F	0	11	1
	G	2	1	11

a. Classification using rbio 3.1 (train1m)

		Target Class		
		C	**F**	**G**
Output Class	C	10	3	0
	F	0	11	1
	G	2	1	11

b. Classification using rbio 3.1 (train1m)

References

[1] R. F. Navea and E. Dadios, "Beta/Alpha power ratio and alpha asymmetry characterization of EEG signals due to musical tone stimulation," in *Project Einstein 2015*, 2015.

[2] P. Manoilov, "EEG eye-blinking artefacts power spectrum analysis," *... Int. Conf. Comput. Syst. ...*, pp. 1–5, 2006.

[3] M. a. Sovierzoski, F. I. M. Argoud, and F. M. De Azevedo, "Identifying eye blinks in EEG signal analysis," *5th Int. Conf. Inf. Technol. Appl. Biomed. ITAB 2008 conjunction with 2nd Int. Symp. Summer Sch. Biomed. Heal. Eng. IS3BHE 2008*, no. 2, pp. 406–409, 2008.

[4] P. Kumari and A. Vaish, "Feature-level fusion of mental task's brain signal for an efficient identification system," *Neural Comput. Appl.*, vol. 27, no. 3, pp. 659–669, 2016.

[5] E. Estrada, H. Nazeran, G. Sierra, F. Ebrahimi, and S. K. Setarehdan, "Wavelet-based EEG denoising for automatic sleep stage classification," *CONIELECOMP 2011 - 21st Int. Conf. Electron. Commun. Comput. Proc.*, pp. 295–298, 2011.

[6] N. Robinson, A. P. Vinod, K. K. Ang, K. P. Tee, and C. T. Guan, "EEG-based classification of fast and slow hand movements using wavelet-CSP algorithm," *IEEE Trans. Biomed. Eng.*, vol. 60, no. 8, pp. 2123–2132, 2013.

[7] R. F. Navea and E. Dadios, "Design and Implementation of a Cascaded Adaptive Neuro-Fuzzy Inference System for Cognitive and Emotional Stress Level Assessment based on Electroencephalograms and Self-Reports," in *HNICEM 2014*, 2014, no. November.

[8] F. R. On, R. Jailani, H. Norhazman, and N. M. Zaini, "Binaural beat effect on brainwaves based on EEG," *Proc. - 2013 IEEE 9th Int. Colloq. Signal Process. its Appl. CSPA 2013*, pp. 339–343, 2013.

[9] P. Anderer, S. J. Roberts, A. Schlgl, G. Gruber, G. Klosch, P. Herrmann, W. Rappelsberger, O. Filz, M. J. Barbanoj, G. Dorffner, and B. Saletu, "Artifact Processing in Computerized Analysis of Sleep EEG - A Review," *Neuropsychobiology*, vol. 40, no. 3, pp. 150–157, 1990.

[10] J. Kim, B. Şen, and et al, "Sleep stage classification based on EEG hilbert-huang transform," *Conf. Proc. ... Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. IEEE Eng. Med. Biol. Soc. Annu. Conf.*, vol. 2014, no. 3, pp. 1–6, 2014.

[11] H. T. Ocbagabir, K. a I. Aboalayon, and M. Faezipour, "Efficient EEG analysis for seizure monitoring in epileptic patients," *9th Annu. Conf. Long Isl. Syst. Appl. Technol. LISAT 2013*, 2013.

[12] A. Subasi and E. Erçelebi, "Classification of EEG signals using neural network and logistic regression," *Comput. Methods Programs Biomed.*, vol. 78, no. 2, pp. 87–99, 2005.

[13] L. Bi, M. Wang, Y. L. Genetu, and F. Aberham, "A shared controller for brain-controlled assistive vehicles," in *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2016.

[14] N. Shinde and K. George, "Brain-controlled driving aid for electric wheelchairs," in *IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2016.

[15] M. Mamun, M. Al-Kadi, and M. Marufuzzaman, "Effectiveness of wavelet denoising on electroencephalogram signals," *J. Appl. Res. Technol.*, vol. 11, no. 1, pp. 156–160, 2013.

[16] P. Kumari and A. Vaish, "Brainwave based user identification system: A pilot study in robotics environment," *Rob. Auton. Syst.*, vol. 65, pp. 15–23, 2015.

[17] P. Kumari and A. Vaish, "Information-Theoretic Measures on Intrinsic Mode Function for the Individual Identification Using EEG Sensors," *IEEE Sens. J.*, vol. 15, no. 9, pp. 4950–4960, 2015.

[18] V. B. Semwal, M. Raj, and G. C. Nandi, "Biometric gait identification based on a multilayer perceptron," *Robot. Auton. Syst.*, vol. 65, pp. 65–75, 2015.

[19] V. B. Semwal, K. Mondal, and G. C. Nandi, "Robust and accurate feature selection for humanoid push recovery and classification: deep learning approach," *Neural Comput. Appl.*, pp. 1–10, 2015.

[20] R. F. Navea and E. Dadios, "Classification of tone stimulated EEG signals using independent components and power spectrum vectors," in *2015 International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM)*, 2015, no. December, pp. 1–5.

[21] S. Ashok and G. Purushotaman, "DWT based Epileptic Seizure Detection from EEG Signals using Naïve Bayes/k-NN Classifiers," *IEEE Access*, vol. 3536, no. c, pp. 1–1, 2016.

[22] I. Belakhdar, W. Kaaniche, R. Djmel, and B. Ouni, "A Comparison Between ANN and SVM Classifier for Drowsiness Detection Based on Single EEG Channel," pp. 443–446, 2016.

[23] A. Turnip, A. I. Simbolon, M. F. Amri, and M. A. Suhendra, "Utilization of EEG-SSVEP method and ANFIS classifier for controlling electronic wheelchair," *Proc. 2015 Int. Conf. Technol. Informatics, Manag. Eng. Environ. TIME-E 2015*, pp. 143–146, 2016.

[24] V. Balasubramaninan, S. Ho, and V. Vovk, "Metaconformal Predictors: Cleassifier Performance Metrics," in *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications*, 2014, pp. 168–169.

[25] N. K. Al-Qazzaz, S. Ali, S. A. Ahmad, M. S. Islam, and M. I. Ariff, "Selection of mother wavelets thresholding methods in denoising multi-channel EEG signals during working memory task," *IECBES 2014, Conf. Proc. - 2014 IEEE Conf. Biomed. Eng. Sci. "Miri, Where Eng. Med. Biol. Humanit. Meet,"* no. December, pp. 214–219, 2015.

[26] M. I. Al-Kadi, M. B. I. Reaz, and M. A. Mohd Ali, "Compatibility of mother wavelet functions with the electroencephalographic signal," in *2012 IEEE-EMBS Conference on Biomedical Engineering and Sciences, IECBES 2012*, 2012, no. December, pp. 113–117.

[27] E. D. Übeyli, "Implementing eigenvector methods/probabilistic neural networks for analysis of EEG signals," *Neural Networks*, vol. 21, no. 9, pp. 1410–1417, 2008.

[28] B. Krose and P. van der Smagt, *An Introduction to Neural Networks*, no. November. The University of Amsterdam, 1996.

[29] V. K. Garg and R. K. Bansal, "Comparison of neural network back propagation algorithms for early detection of sleep disorders," in *International Conference on Advances in Computer Engineering and Applications (ICACEA)*, 2015, pp. 71–75.

[30] L. B. Nguyen, A. V. Nguyen, S. H. Ling, and H. T. Nguyen, "Combining genetic algorithm and Levenberg-Marquardt algorithm in training neural network for hypoglycemia detection using EEG signals," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 5386–5389, 2013.

[31] X. Pan, B. Lee, and C. Zhang, "A comparison of neural network backpropagation algorithms for electricity load forecasting," in *Intelligent Energy Systems (IWIES), 2013 IEEE International Workshop on*, 2013, pp. 22–27.

[32] G. Li and J. Shi, "Applications of Bayesian methods in wind energy conversion systems," *Renew. Energy*, vol. 43, pp. 1–8, 2012.

[33] Z. Yue, Z. Songzheng, and L. Tianshi, "Bayesian regularization BP Neural Network model for predicting oil-gas drilling cost," in *International Conference onBusiness Management and Electronic Information (BMEI)*, 2011.

[34] M. Pal and S. Bandyopadhyay, "Many-objective feature selection for motor imagery EEG signals using differential evolution and support vector machine," in *International Conference on Microelectronics, Computing and Communications*, 2016.

[35] S. A. M. Aris, A. H. Jahidin, and M. N. Taib, "Performance measure of the multi-class classification for the EEG calmness categorization study," in *International Conference on BioSignal Analysis, Processing and Systems*, 2015.

Fuzzy Logic Implementation for MCU on Power Savings and Efficient Irrigation System (MPSEIS) for Smart Farming

Francisco B. Culibrina and Elmer P. Dadios

***Abstract*—Motor speed controller is essential to utilize and maximize the available power. In this paper, the researchers present a self-learning controller for motor speed to be utilize on Three Phase Motor using Variable Frequency Driver (VFD) for irrigation system of Smart Farming using fuzzy logic algorithm developed inside a Micro-Control Unit (MCU) environment or MCU on Power Savings and Efficient Irrigation System (MPSEIS). Motor speed can be reduced and increased using three fuzzy inputs namely, starting process, maintaining process, and stopping process. These fuzzy inputs can be triggered from feedback data of water reservoir level sensor, plant water requirements, and power optimization control. To test the controller's performance, different frequencies using variable frequency driver (VFD) in real time undergoing different water level and power load variations. The whole system is powered by photovoltaic cells, water demand on crops can be quickly and accurately calculated which can be scientific basis for water and power savings for irrigation. For continuous plant production on Smart Farm with independent power supply, new technologies using fuzzy logic were used. The results of experiment showed that the developed controller is reliable, accurate and robust.**

***Keywords*—Fuzzy logic, Irrigation System, MCU, motor speed control, Smart Farm, Three phase motor**

1. Introduction

FOOD security is one of major problems that the world faces today. The effect of global warming coupled with the world population explosion poses a big challenge to solve this issue. The Philippines being a developing country suffered a lot due to lack of food security due to unpredictable weather conditions that destroyed its agricultural products. In addition, due to climate change, the Philippines power/energy generation is compromised that resulted in crops water irrigation problem. As a result, decreasing of agricultural products such as rice, corn, tomato, etc. was sentient by the farmers especially during off-season.

Next crop to rice and corn farming system is tomato, due to continued high demand in the market and increase in consumers. This was considered as one of the most cultivated vegetables having different varieties around the globe [HYPERLINK \l "Pat15" 1]. It has various uses and is attractive to consumers for its health benefits, and it is used as a siding in food presentations and preparations. It is also an important raw material in manufacturing of tomato paste.

From the record of Bureau of Agricultural Statistics (BAS)2] of the Philippines, tomato production increased in 2006 to 2010 from 188.8 thousand Metric Tons (MT) to 204.3 thousand MT which results to an annual growth of 3.87 percent. Increase of area harvested from 17.1 thousand hectares to 17.7 thousand hectares, produced an average yield of 3.08 percent, which grew from 10.26 MT to 11.57 MT per hectares from 2006 to 2010. In 2010, Ilocos is the leading region in production of tomato producing 69.62 thousand MT that contributed about 34 percent of the country's total production. Next is Northern Mindanao contributing 25 percent, followed by Central Luzon and Cavite, Laguna, Batangas, Rizal and Quezon (CALABARZON) region with 10 percent and 9 percent, respectively.

According to World Processing Tomato Council, Figure 1 shows the statistic data of global production and consumption of tomato. It shows that NAFTA, Eu-10, Eu-15, and other parts of Europe has a huge consumption compare with the production. One of the best solutions to address this problem of increasing the demand of tomatoes is an all season crop production. Smart farming

Francisco B. Culibrina and Elmer P. Dadios, De La Salle University, 2401 Taft Avenue, Malate, Manila 1004 Philippines (e-mail: francisco_culibrina@dlsu.edu.ph ; elmer.dadios@dlsu.edu.ph).

or precision farming addresses all season plants/crops production. The focus of smart farming is to use new technology for continuous site-specific plant production [HYPERLINK \l "Mig121" 3]. The continuous monitoring of plant growth is complex and technically challenging. Thus, using fuzzy logic for this new technology, intelligence learning for power saving and efficient irrigation is utilize.

In Smart Farming, the parameters to be monitor are: soil moisture, humidity, temperature and physical appearance of the plants[4].

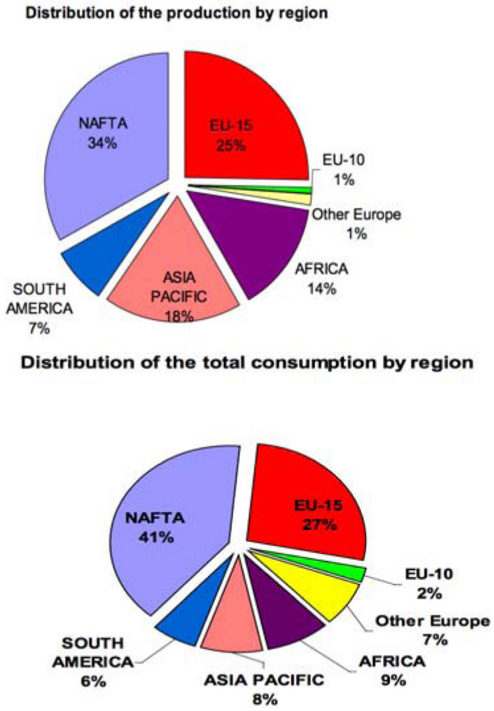


Fig. 1. Global Statistic Data for the Distribution of Tomato Production and Consumption

II. CONTROLLER UNITS FOR POWER SAVINGS AND EFFICIENT IRRIGATION SYSTEM (MPSEIS)

A. Irrigation System

Deep irrigation must be appropriate for tomato and it requires semi-regular rather than light, daily irrigation. One to two inches of soil moisture or one gallon of water each week are the basic requirements of each tomato plants, but more accurately one gallon of water for five days. Infrequent or irregular irrigations for tomato plant results to stress and growth development problems, including blossom end rot and cracked or split fruit [5]. To address these needs, constant water supply must be maintained in spite of limited power source.

Figure 2 shows the power source system and irrigation supply for the Smart farm.

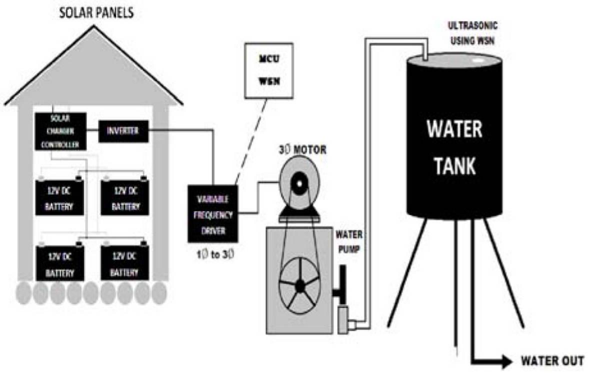


Fig. 2. Overview of Smart Farm Solar and Actuator System Using MCU,WSN, and VFD

Gravitation Cylindrical Water Tank, having a capacity of 1500 liters, was used, located at a total of 58 feet vertical altitude from the plot area. Gravitational tank was used in this research to minimize the use of energy for water pressure. The tank is required to have at least 1000 liters to maintain a minimum of 25 psi to achieve a good performance for automatic sprinkler irrigation and drip irrigation [6].

This research used wireless sensor network for each solenoid valve of drip irrigation and sprinkler irrigation system. In this study, both sprinkler and drip irrigation system shown in Figure 3 has been used. The sprinkler irrigation is used not only to water the plant but also to help lower the temperature and humidity of the plant's environment. The drip irrigation is used to increase the soil moisture content of the plant's surrounding.

Figure 4 shows the general control system for water supply and irrigation.

$$psi = 0.433\ x\ height\ in\ feet \tag{1}$$

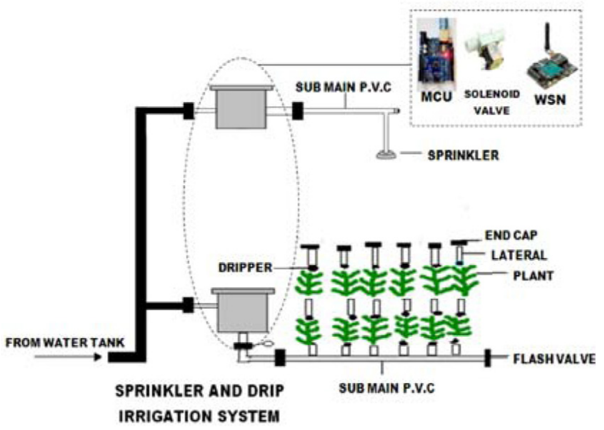


Fig. 3. Sprinkler and Drip Irrigation System with solenoid valve, MCU, and WSN

The system irrigating 500 tomatoes on the farm has a total water consumption of 100 gallons or 400 liters of water every day. The system arrived at a formula shown in equation 1 for water consumption per tomato per liter.

$$\text{Consumption(liters)} = 1\,\text{tomato} \times \left(\frac{1\,\text{gallon}}{3\,\text{days}}\right) \times \left(4\,\frac{\text{liters}}{\text{gallon}}\right) \quad (2)$$

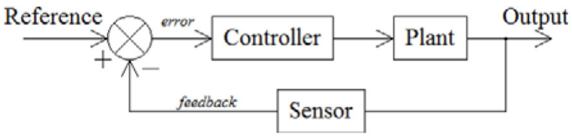


Fig. 4. General Control System

B. Power Requirement

In this paper, the system uses 1 HP three phase induction motor to pump and generate water. This is more efficient compared to a single phase induction motor [7] since the whole system is supplied by 1KW photovoltaic cell (solar panel) and 200 ampere hour storage battery. The challenge of using three phase induction motor was encountered and it was addressed by using Variable Frequency Drive (VFD) and Micro Control Unit (MCU). The MCU were programmed by implementing Fuzzy Logic Control.

1. Variable Frequency Drive (VFD)

VFD is a device used to control the AC motor speed and torque by varying its input frequency and voltage [HYPERLINK \l "Dad12" 5] [6]. The VFD is used to drive an electric motor by varying the frequency and voltage supply. To prevent high current requirements in starting up a 1 HP three phase motor, the VFD is programmed for gradual increase of frequency from 0Hz to 60Hz within 5 seconds. Since frequency are directly related to the motor speed (rpm) [HYPERLINK \l "Eng10" 7], starting-up of three phase induction motor requires only small amount of current compared to direct single phase motor. Experiment data can be seen in Figure 10.

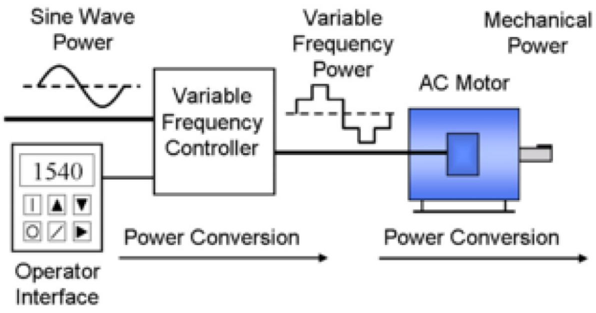


Fig. 5. Basic operation of VFD and 3 Phase Motor

2. Three Phase Induction Motor

Figure 5 shows the actual 1 HP three phase induction motor used in this research that runs on a three phase AC supply. The advantage of this setup is that the construction is simple and rugged, reliable, highly efficient, has excellent power factor, economical, and requires minimum maintenance [8].

III. Processes Involved in Formulating the Fuzzy-MPSEIS

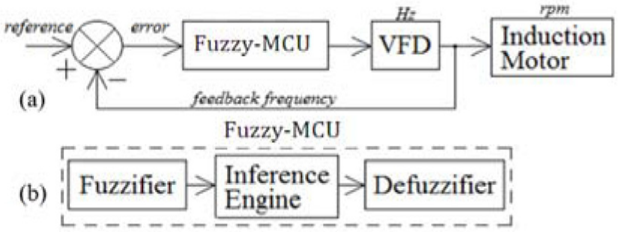


Fig. 6. Components of the fuzzy logic - MCU controller

A. The Fuzzy Logic-MCU Process Flow

The architecture of the fuzzy logic controller implemented for power savings and efficient irrigation system using a Micro-Controller Unit (MCU) can be seen in Figure 7. The starting point (SP) served as the reference value of the induction motor speed. Another input, process value (PV), is obtained as the feedback value of the variable frequency drive (VFD). SP and PV will be used to solve error and rate of error. Figure 7 shows an illustration of the fuzzy control used. In this setup, the micro-controller unit (MCU) served as the main controller where the fuzzy logic algorithm was programmed. The fuzzy control in this research consists of a fuzzifier, an inference engine, and a defuzzifier[HYPERLINK \l "Mam74" 9 , HYPERLINK \l "RLa99" 10].

B. The MCU Programming

In this study, there are two inputs and one output of the system. The inputs are the water level and energy available, and the output is the motor speed. The water level has three membership functions which are L, M, and H. L stands for low, M stands for Medium and H stands for high level. For energy available there are four membership functions which are VL, L, M, and H.

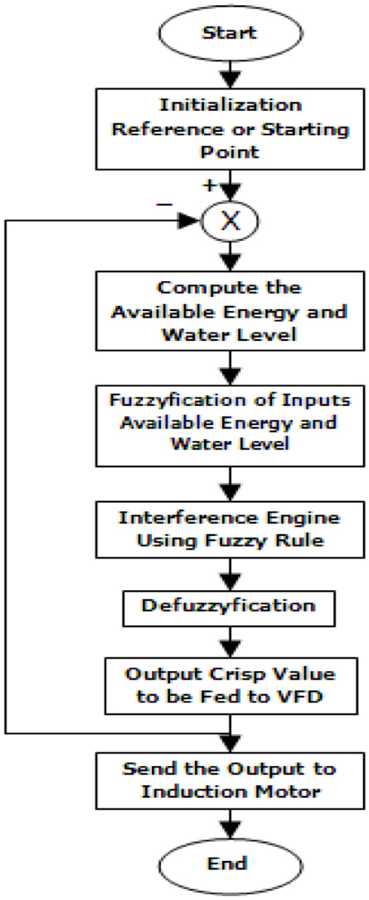
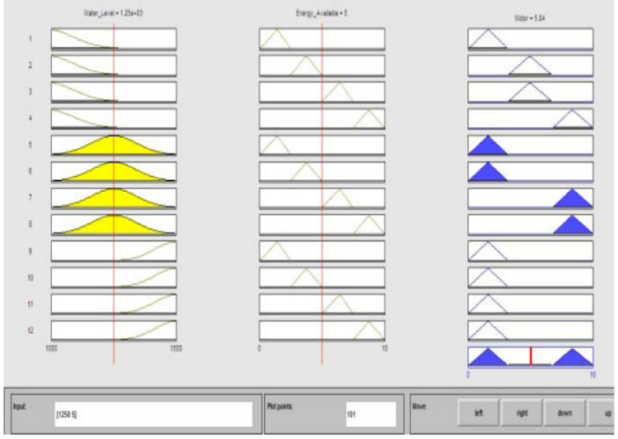

Start
Initialization Reference or Starting Point
+
–
X
Compute the Available Energy and Water Level
Fuzzyfication of Inputs Available Energy and Water Level
Interference Engine Using Fuzzy Rule
Defuzzyfication
Output Crisp Value to be Fed to VFD
Send the Output to Induction Motor
End

Fig. 7. Fuzzy Control Flow Chart Diagram

VL indicates for very low, L stands for low, M stands for medium and H stands for high. The output motor speed has three membership functions such as Stop for stopping, M for moderate running and H for High speed running.



Input: [1250 5]

Plot points: 101

Move:

Fig. 8. Fuzzy Logic Rule and Membership Function

TABLE I
Fuzzy Rule

Water Level	Available Energy	Motor Speed
L	VL	Stop
L	L	M
L	M	M
L	H	H
M	VL	Stop
M	L	Stop
M	M	H
M	H	H
H	VL	Stop
H	L	Stop
H	M	Stop
H	H	Stop

IV. Experiment Results

The information given in Table 1, Figure 7 and Figure 8 were implemented in C language program. The response of motor controller is shown in Figure 9. This indicates that the energy consumed for the whole system is very minimal. *The output energy consumption for the irrigation system to fill the tank is 3.73kW-hr/day. To maintain the 1500 liters of water as mentioned earlier, the energy requirements for the system is 1.24kW-hr/day.*

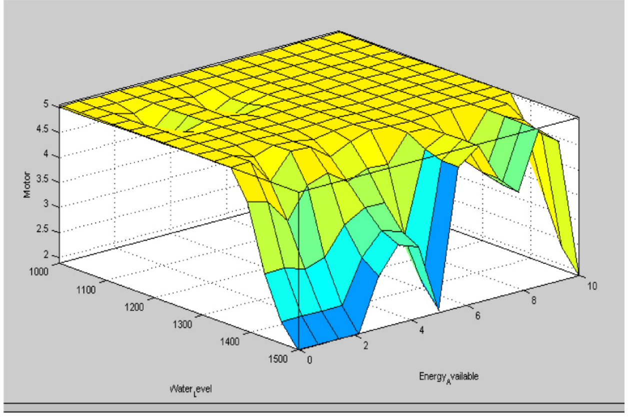


Fig. 9. Fuzzy Surface output for Water level, Energy available and Motor

V. Discussions and Analysis of Results

Figure 10 shows the comparison of the results of the current consumption for the proposed MPSEIS against the conventional system. Note that the conventional system uses 1HP single phase induction motor. It was observed that the conventional system required high current for starting process. This also gave a very high average current consumption of 8.1 amperes. This problem was solved using three-phase induction motor with variable frequency drive (VFD) implementing the fuzzy logic controller that maximized the percent energy savings. It is possible even if the supply coming from photovoltaic cell (solar panel), the system can manipulate the starting revolution per minute (rpm) of motor by gradual increasing the frequency (Hz) from 0Hz to 60Hz induced in the motor, since the rpm of motor is directly proportional to frequency. The current requirements resulted into a very minimal value. It was also noted that the current consumption of MPSIES had an average of 2.0 amperes, while the conventional system had an average of 8.1 amperes, which can be seen on Table II. The graph also showed that the out

MPSIE are linear while conventional system are non-linear.

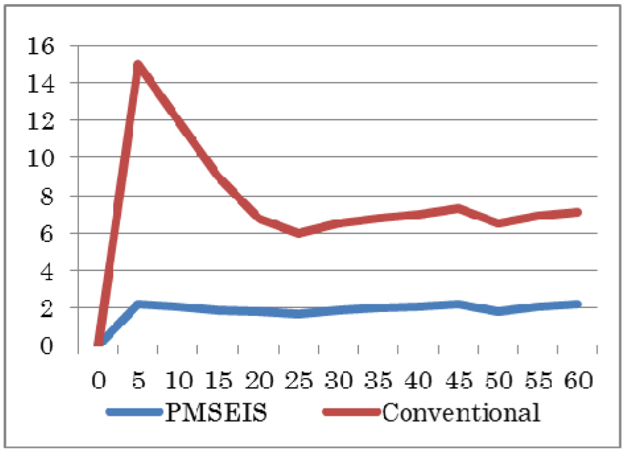


Fig. 10. Line graph of current consumption for PMSEIS and Convention

TABLE II
% Savings and Current Consumption of MPSEIS and Conventional System

Time in seconds	0	5	10	15	20	25	30	35	40	45	50	55	60
PMSEIS	0	2.2	2.1	1.9	1.8	1.7	1.9	2	2.1	2.2	1.8	2.1	2.2
Conventional	0	15	12	9	6.8	6	6.5	6.8	7	7.3	6.5	6.9	7.1
% Savings	0	85.3	82.5	78.9	73.5	71.7	70.8	70.6	70.0	69.9	72.3	69.6	69.0

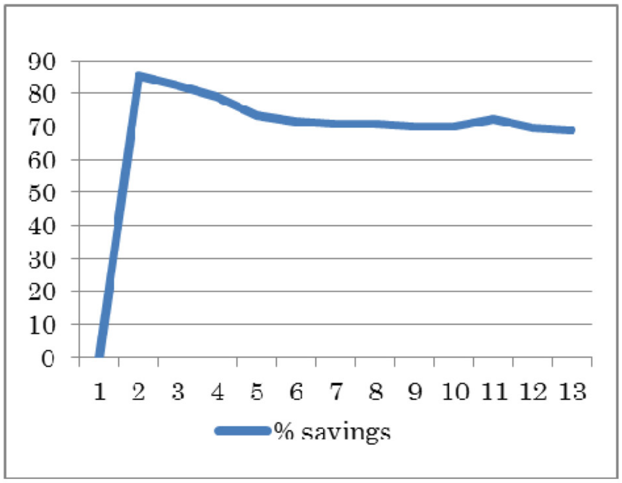


Fig. 11. Percentage Savings of Current

Figure 11 shows the percentage savings of current where calculated based on the Equation 3. The average current savings was computed equal to 73.7% compare from the conventional system.

$$\%\,\text{savings} = \frac{\text{con.current} - \text{MPSEIS Current}}{\text{Con.Current}} \times 100 \qquad (3)$$

VI. Conclusion and Recommendations

Using VFD and MCU with Fuzzy Logic-Based Motor Speed Controller into Irrigation System for Smart Farming was proven to be successful. Based on the experiment results, it is significantly better and reliable compared to the conventional controller in terms of load variations handling.

For enhancing the capabilities of MCU, fuzzy logic was appreciated for providing intelligent automation system process. The result of the developed controller showed that it is reliable, accurate and robust by providing a percentage savings of current usage which is equal to 73.7 percent.

Acknowledgements

The authors would like to acknowledge University of Rizal System–Morong, De La Salle University–Manila, and the Engineering Research and Development for Technology (ERDT) of the Department of Science and Technology (DOST) for the research support and funding.

REFERENCES:

www.engineeringtoolbox.com

[1] ElectroTechnik. (2012) *electrotechnik*. [Online]. www.electrotechnik.net

[2] E. H. Mamdami, “Application of fuzzy algorithm for the control of a dynamic plant,” *IEEE Proc.*, vol. 121, pp. 1585–1588, 1974.

[3] R. Langari, “Past, present and future fuzzy cotrol; a case for application of fuzzy logic in hierarchical control,” in *18th Int’l Conf. of the North American Source*, 1999, pp. 760–765.

[4] Electrical Engineering. (2012, April) *electrical-engineering portal*. [Online]. www.eletrical-engineering-portal.com

[5] Arduino. (2000) arduino.cc/en/Tutorial/*ArduinoToBreadboard*. [Online]. www.arduino.cc

[6] [Online]. www.veggiegardener.com

[7] Veggiegardener, “5 Tips to Enhance Tomato Health, Growth and Taste,” *Growing Tomato*, March 2015.

[8] [Online]. www.irrigationtutorials.com

[9] Jess Stryker, “Irrigation Design - Landscape Sprinkler System Design Tutorial,” *Irrigation Tutorial*, June 2013.

The Contributors

Carlo Noel Ochotorena received B.S. and M.S. degrees in Electronics and Communications Engineering from De La Salle University, Manila, Philippines in 2009 and 2012, respectively.

Since 2013, he has been a professor with the Electronics and Communications Engineering at De La Salle University, Manila Philippines. He is also currently a Doctoral student at the Tokyo Institute of Technology. His research interests include image processing, compressive sensing, and computational photography.

Cecille Adrianne Ochotorena received the B.S. and M.S. degrees in Electronics and Communications Engineering from De La Salle University, Manila, Philippines in 2009 and 2012, respectively.

From 2013 to 2014, she was a lecturer with the Computer Technology Department at De La Salle University, Manila Philippines. She is currently pursuing her Doctoral studies in Electronics and Communications Engineering at De La Salle University. Her research interests include computer vision and computational photography.

Yukihiko Yamashita was born in 1960 in Kanagawa Prefecture, Japan. He received the B.E., the M.E., and the Dr. Eng. degrees from Tokyo Institute of Technology in 1983, 1985, and 1993, respectively.

From 1985 to 1988 he was with the Japan Atomic Energy Research Institute. From 1988 to 1989 he was with the ISAC corporation. In 1989 he joined the faculty of the Tokyo Institute of Technology, where he is now an associate professor of the Course of Engineering.

His research interests include pattern recognition and image processing.

He received a Paper Award in 1993 from the Institute of Electronics, Information, and Communication Engineers of Japan (IEICE).

Dr. Yamashita is a member of the Institute of Electrical and Electronics Engineers, the Institute of Electronics, Information, and Communication Engineers of Japan, and Information Processing Society of Japan.

Alberto S. Bañacia is currently an assistant professor in the Department of Electrical & Electronics Engineering at the University of San Carlos in Cebu City, Philippines. He received his BSc in Electronics and Communications Engineering from the same university in 1990 and his Master of Information Engineering from Mie University, Japan, in 2004. He has served USC in various capacity as Department Chairman, Laboratory head, and Academe-Industry Linkage Coordinator.

A professional Electronics engineer (PECE), he is an active member of the Institute of Electronics Engineers of the Philippines (IECEP) Cebu chapter and a member of the different societies affiliated with the Institute of Electrical and Electronics Engineers (IEEE) namely IEEE Signal Processing Society, IEEE Communications Society, and IEEE Education Society.

His research interests are in the areas of image processing (face detection, tracking, and recognition), modulation, source and channel coding, wireless communications systems on software-defined radio platforms, cognitive radio, and TV white space.

He is currently finishing his PhD at the Mapua Institute of Technology in Intramuros, Manila.

Joy Alinda Madamba graduated in BS Computer Engineering and MS Electrical Engineering Major in Microelectronics from the University of the Philippines–Diliman in 2000 and 2005, respectively. She is currently a PhD student in Electrical Engineering in UP Diliman. She is also an Assistant Professor in the Electrical and Electronics Engineering Institute (EEEI) of UP Diliman. Her research interests are computer architecture, multicore processors, and digital IC design.

Francis Joseph Seriña, hailing from the University of the Philippines Diliman, graduated with a Bachelor's Degree in Computer Engineering in 2008. He has then worked as a software developer, game developer, and educator. He recently took his Masters in Computer Science in DigiPen Institute of Technology, specializing in Computer Vision. Currently, he works as a Software Engineer for Autodesk Inc.

Geoffrey L. Abulencia was born in Manila, Philippines in 1987. He received his B.S. degree in Electronics and Communications Engineering (ECE) from the Pamantasan ng Lungsod ng Maynila, in 2009, and his M.S. degree in ECE from the De La Salle University – Manila, in 2014. He previously worked as a researcher at the Department of Science and Technology (DOST) and currently working as a lecturer at DLSU-Manila under ECE department.

Alexander C. Abad is a graduate of BS Electronics and Communications Engineering at St. Louis University–Baguio City. He finished his MS in Electronics and Communications Engineering at De La Salle University –Manila and currently taking up his PhD in ECE at DLSU. His field of interest includes robotics, machine intelligence, machine vision, mixed-signal electronics, and IC design.

Krister Ian Daniel Z. Roquel is currently taking his PhD in Civil Engineering at De La Salle University Manila. He worked at De La Salle University – Manila as a part-time lecturer in the Civil Engineering Department shortly after earning his Masters in Civil Engineering at the same University. He specializes in Transportation Engineering and focuses his research on discrete choice modeling and transportation economics.

Alexis M. Fillone is a full professor of the Transportation Engineering Division, Civil Engineering Department of De La Salle University, Manila. He earned his PhD in Urban and Regional Planning at the School of Urban and Regional Planning, University of the Philippines, Diliman and his Master of Engineering (Transportation) at the Asian Institute of Technology, Bangkok, Thailand. Has has 25 years of experience as a teacher in the field of transportation planning and engineering and has more than 15 years of experience in transport research focusing on travel behavior, urban transportation planning, and traffic impact studies. He has published several articles in abstracted journals and is a co-author in two published books.

Roy Francis Navea was born in Biñan, Laguna in 1981. He received the B.S. degree in Electronics and Communications Engineering (ECE) from the Mapua Institute of Technology, Intramuros, Manila in 2003, and the M.S. degree in ECE from the De La Salle University (DLSU), Manila in 2012. He is a PhD in ECE candidate in DLSU with his dissertation focusing on the analysis of musically stimulated brain wave signals.

Currently, he is an Assistant Professor in the Electronics and Communications Engineering Department of DLSU. He is based in its Laguna campus, the De La Salle University – Science and Technology Campus (DLSU-STC). His research interests include digital signal processing, artificial/computational intelligence, bioinformatics, and wavelet analysis.

Francisco Bolante Culibrina born on October 4, 1974 at Subay, Cardona, Rizal, only child of Mr. Isagani V. Culibrina and Mrs. Josephina B. Bolante. He graduated three years Computer Technology on March 19, 1994 at Rizal Technological University. He continued his studying as Bachelor of Science in Electronics and Communications Engineering (BS-ECE) and graduated on April 11, 1997 at the same university.

After graduation, he work as a free lancer technician and installer. He worked at Crystal Semiconductors Inc. at Mandaluyong City as Quality Control/Quality Assurance (QC/QA) Engineer on June 18, 1999. While working he attend review for ECE, he took Professional Regulation Commission (PRC) and passed the licensure examination for ECE on November 4–5, 2000. After passing the licensure exam, he promoted as QC/QA Supervisor on December 5, 2000. He resigned at semiconductor company and became a Engineering Instructor at University of Rizal System on June 16, 2001. Became Network Engineer at Bayan Telecommunication on June 18, 2007 and returned as engineering lecturer at the same year and university.

He got married to Vanesa P. Bautista on April 7, 2001 at First Assembly of God Church, Angono, Rizal. He had five children, named: Francisco Jr., Frinsford, Frainrave, Farahvien and Feahfaith.

March 9, 2010, he passed PRC examination for Professional Electronics Engineer (PECE). After passing PECE examination, he became a consultant to various electronics and semi-conductors from Metro Manila and nearby province. October 21, 2013, he graduated Master of Science in Electrical Engineering major in Electronics Engineering (MS-EE) at Technological University of the Philippines (TUP)- Manila. Became PhD-ECE student at De La Salle University (DLSU) – Manila as DOST/ERDT scholar on May 1, 2014.

Currently, he is Assistant Professor at College of Engineering, University of Rizal System – Morong. Aside from being consultant from electronics and semiconductors company, he is also full time researcher and work for his dissertation research at De La Salle University in Smart Farming project, funded by Philippine government through DOST as well as by USAID STRIDE.

Elmer Dadios

Affiliation: University Fellow and Professor, De La Salle University President, NEURONEMECH, INC

Brief Biographical History:

1996 Received Doctor of Philosophy from Loughborough University

1997 Exchange Scientist, Japan Society for the Promotion of Science, Tokyo Institute of Technology

1998–1999 Director, Engineering Graduate School, De La Salle University

2003–2004 Director, School of Engineering, De La Salle University, Science Technology Complex

2003-2015 General Chair, HNICEM International Conference

Main Works:

- "Fuzzy Logic Book – Controls, Concepts, Theories and Applications," ISBN: 978-95351-0396-7, 2012.
- "Fuzzy Logic Book – Algorithms, Techniques and Implementations," ISBN: 978-953-510393-6, 2012.
- "Fuzzy Logic Book – Emerging Technologies and Applications," ISBN: 978-953-51-03370, 2012.
- Research interests includes; Robotics, Mechatronics, Automation, and Computational Intelligent System.

Membership in Academic Societies:

- Senior Member, The Institute of Electrical and Electronics Engineers (IEEE)
- Founder and Currently the Chair, IEEE Computational Intelligence Society, Philippines
- IEEE Region 10 Executive Committee Member
- IEEE Rio Awards and Recognition Committee Chair
- Founder and DOST President, The Mechatronics and Robotics Society of the Philippines

Guidelines for Contributors

1. The Journal on Computational Innovations and Engineering Applications (JCIEA) aims to promote the development of new and creative ideas on the use of technology in solving problems in the field of computational applications, computational intelligence, electronics and information and communications technology (ICT), manufacturing engineering, energy and environment, robotics, control and automation, and all their related fields. Manuscript submissions should, therefore, be in pursuit of the same goal and within the related fields.
2. JCIEA only accepts manuscripts written in English. The responsibility for copyediting manuscripts, as well as obtaining reproduction permissions for the use of graphics and other materials from their references, will fall on the author.
3. Authors must also remember to cite all references and ensure that their paper submission has not been previously published or is undergoing peer review for another publication.
4. Manuscripts should include a unique title, an abstract, some keywords, an introduction and discussion of the study, a presentation and discussion of results, and a conclusion. Authors may also include an acknowledgement of funding organizations or consultants, if needed.
5. Manuscripts may be sent to *jciea.dlsu@gmail.com* or *jciea@dlsu.edu.ph* as either an MS Word file (*.doc or *.docx) or a LaTeX file (*.tex), including its supporting files.
6. Manuscripts in either file format should have the following features:

- Single-spaced, two-column format with 1-inch margin on all sides on letter-sized template
- Font to be used is Times New Roman, size 11
- Graphs (*.eps, *.svg), tables (*.csv), and images (*.jpg, *.png) should be saved and sent apart from the MS Word file.
- Citations and references should be submitted in IEEE or APA format.
- Submission of these references in a BibTeX format is preferred.

7. Manuscript should be eight to twelve (8–12) pages long, including all figures, tables, and references. Manuscripts exceeding the 12-page limit will require permissions from the editors.
8. Authors must include their full names and affiliations in the manuscript. They may include a 150- to 200-word biography to be included in the back portion of the journal.