Volume 6, No. 5; October 2018

Advances in Image And Video Processing

ISSN: 2054-7412

Society for Science and Education - United Kingdom

TABLE OF CONTENTS

EDITORIAL ADVISORY BOARD

DISCLAIMER

All the contributions are published in good faith and intentions to promote and encourage research activities around the globe. The contributions are property of their respective authors/owners and the journal is not responsible for any content that hurts someone’s views or feelings etc.

Society for Science and Education
United Kingdom

AIVP Advances in Image and Video Processing

Volume 6, Issue 5
ISSN 2054-7412

Flower Image Retrieval Using LBP, Wavelet Moments, Gabor Wavelet Features and Multiple Distance Functions

Zannatul Azme, Khadiza Sultana Happy, Mohammad Farhad Bulbul*
Department of Mathematics,
Jessore University of Science and Technology, Jessore, Bangladesh
farhad@just.edu.bd

ABSTRACT

Flower image retrieval is a significant and challenging problem in content-based image retrieval. In this paper, we propose a content-based method for retrieving flower images of specified specie from a database of flower images of various species. Firstly, we use wavelet moment, Gabor wavelet and Local Binary Pattern (LBP) independently to characterize all flower images in the database. Secondly, we represent a query flower image with Gabor wavelet, wavelet moment, and LBP individually and search images in the database analogous to the query image. The retrieval is accomplished through calculating similarities between the query image and the database images by employing a set of distance functions. Experimental evaluation of the approach reveals that the Gabor wavelet achieves superiority over the wavelet moment and LBP considerably. It is also indicated that the retrieval outcome can be improved through concatenating the Gabor wavelet, wavelet moment, and LBP features rather than utilizing them individually.

1 Introduction

At present world, with the availability of image capturing devices such as digital cameras, mobile cameras, webcams, video cameras, image scanners, the size of digital image collection is increasing at a great rate. A computer system for image searching, browsing and retrieving from a large database of digital images is called an image retrieval system. It is a tool, required by users from various domains, including remote sensing, medicine, crime prevention, architecture, publishing, fashion, etc. From this point of view, there have been developed many prevalent image retrieval systems. There are two frameworks of image retrieval such as text-based image retrieval and content-based Image Retrieval. The text-based approach started since the early 1960s. In text-based image retrieval system, for finding similar types of images it is needed to search by the accurate textual description, filenames, caption, keywords etc. Because the images are annotated in the database by textual description, filenames, caption, keywords etc. But text- based image retrieval is oldest image searching system. There are many several problems of text- based image retrieval methods. Firstly, annotation is never complete because it depends on the goal of the annotation, some mistakes like spelling error, spelling difference (US vs.UK), weird abbreviation (particularly medical) etc. The use of keywords is not only complex but also insufficient to represent the content by the size of image databases growing etc. The keywords are not unique for one kind of searching. The problems of image queries by text-based approaches cannot be described at all, but tap into the visual features of images .For example, a query for all the images in the database with "classroom" in it will good results if we annotate all the images containing classroom but for the same annotations, a specific search for images with a chair, table,

DOI: 10.14738/aivp.65.5327
Publication Date: 4th November, 2018
URL: http://dx.doi.org/10.14738/aivp.65.5327

computer or student in tit will fail. Another problem with text based approach is the lack of uniform textual descriptions of such attributes like color, shape, texture etc. So users are not satisfied with the conventional text-based retrieval methods. CBIR that means Content-based Image Retrieval is a technique that helps to access and arrange the digital images from a giant database by using visual contents of the images i.e. the features of images. The word "Content- based" express that the query will be analyzed by the actual contents of the image other than the metadata such as tags, keywords or descriptions linked with the image. Here the 'content' means colors and textures information that can be extracted from the image. The focus of CBIR is to avoid the use of textual descriptions. CBIR is such a process that will ease the data handling and the user can easily access the data. Basically, the technique tries to retrieve images similar to a user-defined specification or pattern (e.g., shape sketch, image example). So in CBIR, retrieving of an image based on focusing at their contents like colors, textures, shapes etc. The CBIR process has gain utmost importance because most web-based image search engines purely depend on metadata and this produces a lot of false identification in the search results. For these issues, CBIR has become an active and fast moving research sector since the 1990s. There are several advantages of CBIR in real time situation. Two major research communities, database management and computer vision have become a very active area of research on CBIR. CBIR technique is easy, effective, and less expensive. It takes less time to find all those related images. All over that, present world image retrieval mainly focused on Content-based image retrieval system. The possibility of an automatic retrieval process is one of the main advantages of the CBIR approach, instead of the regular keyword-based approach, which usually time-consuming and very laborious. However, CBIR technology has been used in several applications such as fingerprint identification, biodiversity information system678s, digital libraries, Crime prevention, Security Check, Medical Diagnosis, Intellectual Property, Art collection, Medical Image Databases, Scientific Databases, General Image collections for licensing, Architectural and Engineering design, Fashion and publishing, Military, photograph archives, face finding, geographical information and remote sensing systems and so on.

Flower image retrieval system is one type content-based image retrieval system and it is very significant for computer-aided plant species recognition. Flower image databases and collections can be enormous in size, containing hundreds, thousands or even millions of flower images. Difficulties for flower image retrieval are as follows there complex backgrounds, variations of brightness and intensity under different natural illuminations, shadow effects of the surrounding, diversity of flowers as different appearances, different camera angles, different sizes, and different resolution, etc. As a result, flower image retrieval is a significant and challenging problem in content-based image retrieval. We had systematic and overall researches on flower images, feature fusion, regional segmentation and feature extractions etc. [1], [2]. There are about 250000 named kinds of flowering plants. But many plant species that have not been classified and named. Plant organization and identification is a very ancient area. A major development can be projected if the plant identification can be carried out by a computer automatically with the aid of image processing and computer vision techniques, and many data management techniques.

Many researchers have been worked at flower image retrieval system. For example, in 2008, Maria-Elena Nilsback and Andrew Zisserman [2] introduced a 103 class flower dataset. They compute four different features for the flowers, each describing different aspects, namely the local shape/texture, the shape of the boundary, the overall spatial distribution of petals, and the color. They combined the features using a multiple kernel framework with an SVM classifier. The weights for each class are learnt using the method of Varma and Ray [3], which has achieved state of the art performance on other large dataset, such as Caltech 101/256. Their dataset had a similar challenge in the number of classes, but with the added difficulty of large between class similarity and small within class similarity. Their results showed that learning the optimum kernel combination of multiple features vastly improves the performance, from 55.1% for the best single feature to 72.8% for the combination of all features. After two years, Xiao Ke et al. [4] discussed on recent developments in content-based flower image retrieval. They studied repetitive image detection algorithm based on canny edge to filter repetitive images in flower database and adaptive threshold segmentation algorithm based on 2RGB mixed color to segment flower image. They showed this segmentation results of these flowers are not sampling that is these flowers was not completely extracted from entire images. Again, they tested adaptive threshold segmentation algorithm based on 2RGB mixed-color model used in these part of testing. These types of tested flowers were similar to other kinds of flowers. Finally to test the flower image retrieval system, they used HSV histogram, color moments, GLCM, Gabor texture, weighted invariant moments and edge LBP features. They also tested the retrieval results by using the strategy of multi-feature fusions. They obtained very good retrieval results of flower image retrieval system based on multi-feature fusions. They used total 15 categories flower images, under the premise of returning 20 most similar images for each test image, the obviousness of 12 categories' flowers passed 80% and other three kinds of flowers passed 70%.

In the same year, Hsu et al. [5] presented an interactive system for recognizing flower images used by digital cameras. They proposed a system which provides a simple user interface which allows each user to draw a rectangular bounding window containing the interested flower region. They developed to find the flower boundary as accurately as possible by a boundary tracing algorithm. Along the color and shape features of the total flower region, the color and shape features of the pistil/stamen area are also removed to represent the flower characteristics in a more appropriate currency. They showed their experiments conducted on two different flower image databases consisting of 24 species and 102 species. Finally, they proposed system outperforms other approaches in terms of the recognition rate than the methods proposed by Hong et al. [6], Zou and Nagy [7], and Saitoh et al. [8].

Several years later, HU et al. [9] developed another content-based flower image retrieval system. They researched on flower image retrieval algorithm based on saliency map. They avoided traditional image segmentation. Firstly, they improved and used Itti's visual attention model, and extracted the color and LBP texture features using the saliency map. They gained more effective retrieval results than the other results based on the saliency map extracted through Itti’s model.

In this paper, we propose a system for content-based flower image retrieval. For a given query flower image, our system goals to return a set of representative images in which similar types of flowers look like. We use some descriptors such as wavelet moments, Gabor wavelet and LBP to represent the query image as well as the database images. A set of distance functions is utilized to measure image similarity. In this work, we build the system by using the Gabor wavelet, wavelet moments, and LBP descriptors individually and collectively.

The rest of this paper is organized as follows. The proposed approach is presented in Section 2. The experimental results are demonstrated in Section 3. Finally, Section 4 contains a brief conclusion of this work.

2 Our approach

In this section, we discuss about our approach for retrieving flower images. We firstly discuss on the feature extraction from query and database flower images and then about the distance functions used for finding out the similarity between the query image and the database images.

Figure 1 represents our architectural overview of the proposed framework.

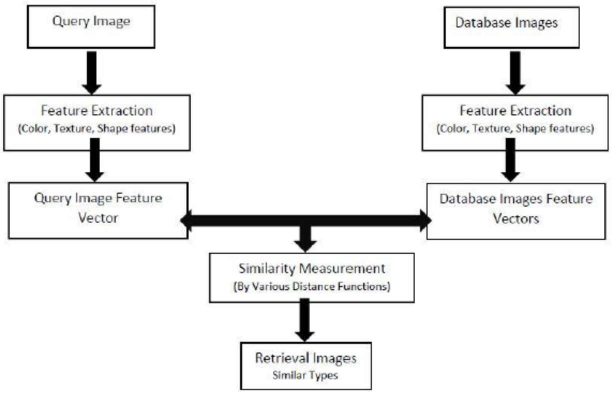

Query Image
Database Images
Feature Extraction
(Color, Texture, Shape features)
Feature Extraction
(Color, Texture, Shape features)
Query Image Feature Vector
Database Images Feature Vectors
Similarity Measurement
(By Various Distance Functions)
Retrieval Images
Similar Types

Figure 1: Architectural overview of the proposed framework

2.1 Feature Extraction

Our work uses wavelet moment, Gabor wavelet and Local Binary Pattern (LBP) features to describe a flower image. This section discusses those features briefly.

Wavelet Transform: A wavelet is a small wave. Wavelet transform is the representation of a function by wavelets. It is used for extraction of texture feature. Wavelets are defined as small waves of different frequency having limited [10]. Wavelet transform is broadly used in computer vision as an image processing technique for object identification and classification. Application of Discrete Wavelet Transform (DWT) as an image processing system produces the revolution values called wavelet coefficient. A common method of feature extraction from wavelet transformation is the calculation of coefficient distribution over elected mother of wavelet. Naturally, the actualization of DWT uses the mother of wavelet such as Haar, daubechies, Coiflet, Meyer, Morlet and Mexican Hat. However, this study only considers the Coiflet mother wavelet. The fundamental idea behind wavelets is to analyze signal according to scale. The basic idea of DWT is to provide the time-frequency representation. The 2D-DWT represents an image in terms of a set of shifted and dilated wavelet functions ψLC ,ψCL ,ψCC and scaling function ϕLL that form an orthonormal basis for $L2(R2)$. Given J-scale DWT as image $(s,)$ of N×N is decomposed as

$$x(s,t) = \sum_{k,i=0}^{N_J-1} U_{J,k,i}\, \phi_{J,k,i}^{LL}(s,t) + \sum_{B\in B} \sum_{j=1} \sum_{k,i=0}^{N-1} W_{j,k,i}^{B} \psi_{j,k,i}^{B}(s,t) \qquad (1)$$

$$\begin{aligned} \text{Where, } \phi_{j,k,i}^{LL}(s,t) &\equiv 2^{-j/2}\, \phi(2^{-j}s-k, 2^{-j}t-i), \psi_{j,k,i}^{B}(s,t), \psi_{j,k,i}^{B}(s,t) \\ &\equiv 2^{-j/2}\, \psi^{B}(2^{-j}s-k, 2^{-j}t-i),\ B \in \boldsymbol{B}, \boldsymbol{B} \end{aligned} \qquad (2)$$

{LC, CL, CC} and$Nj=N2j/$. In this paper LC, CL, and CC are called wavelet or DWT sub-bands. $Uj,k,i=\iint x(s,t)\phi J,k,ids\ dt$ is a scaling coefficient and $Wj,k,iB=\iint x(s,t)\psi j,k,iBds\ dt$ denotes the (k, i)th wavelet coefficient in scale j and sun-band B [11]. Figure 2 represents the scaling model in wavelet transform.

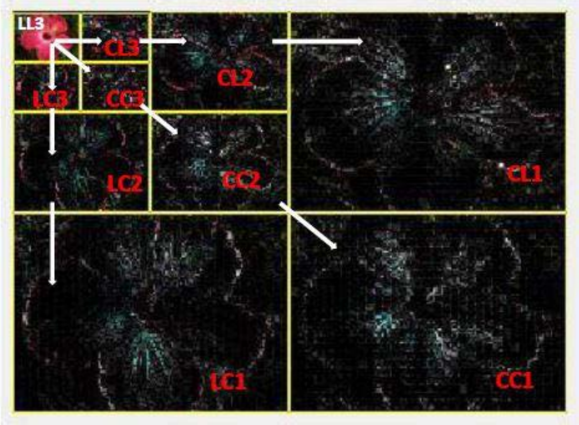


Figure 2: Joint spatial and frequency representation of a 2-D three-scale DWT

We use the image processing method that have been applied on the feature vector extraction. Applying the wavelet transform to the image with a 3-level decomposition, the standard deviation and mean of the transform coefficient CC1 is used to form the feature vector.

Gabor Wavelet: Let an image be *I* (*u*, *v*) with $m \times n$ size. Then its discrete Gabor wavelet transform is given by a convolution [12]:

$$G_{pq}(u,v) = \sum_{x}\sum_{y} I(u-x, v-y)\psi^{*}_{pq}(x,y) \tag{3}$$

where,

x and *y* are the filter mask size variables

ψ_{pq*} is the complex conjugate of ψ_{pq}.

ψ_{pq} is a class of self-similar functions generated from dilation and rotation of the following mother wavelet:

$$\psi(s,t) = \frac{1}{2\pi\sigma_s\sigma_t}\exp\left[-\frac{1}{2}\left(\frac{s^2}{\sigma_s^2}+\frac{t^2}{\sigma_t^2}\right)\right].\exp(j2\pi Vs) \tag{4}$$

Here,

V = modulation frequency.

The self-similar Gabor wavelets are obtained through the generating function:

$$\psi_{pq}(s,t) = a^{-p}\psi(\tilde{s},\tilde{t}) \tag{5}$$

Here,

, respectively denotes the scale and orientation of the wavelet with $p=0$,1,...$P-1$,$q=0$,1,...,$Q-1$

And

$$\tilde{s} = a^{-p}(s\cos\theta + t\sin\theta) \tag{6}$$

$$\tilde{t} = a^{-p}(-s\sin\theta + t\cos\theta) \quad , \; a > 1 \; and \; \theta = \frac{q\pi}{Q} \tag{7}$$

Local binary patterns (LBP): Local binary pattern (LBP) is a type of visual descriptor used for pattern classification in the area of computer vision [13]. LBP features encode local texture information, which we can use for tasks such as classification, detection, and recognition. LBP is a simple yet very efficient texture operator. The LBP operator typically works in 3× 3 pixel blocks. All working block are thresholded by the center pixel. Its weighted by powers of 2 and then mixed to label the center pixel. These labels are the pixels of a picture by thresholding the neighborhood of every pixel then the result as a binary number are conceive. The operator can also be extended to neighborhood with various sizes [14]. Let us consider a neighborhood denoted by(P,Q), here P represents the sampling points and Q represents the radius. These sampling points lie around the center pixel $(U,)$ and at coordinates are

$$(U_m, V_m) = \left(U + Q\cos\left({}^{2\pi m}/_{P}\right), V - Q\sin\left({}^{2\pi m}/_{P}\right)\right) \qquad (8)$$

Now the LBP label for pixel $(U,)$ can be calculated as follows

$$LBP_{P,Q}(U,V) = \sum_{m=1}^{P-1} T\big(f(U_m, V_m) - f(U,V)\big).\,2^{m-1} \qquad (9)$$

Where (U)=1 if $(U{\geq}0)$ and $T(U)$=0 if U<0 [15].

Accordingly pixel value is summed for the LBP number of this texture unit. LBP technique is gray scale invariant and can be easily joined with a simple contrast amount by adding for each neighborhood the difference of the average gray level of those pixels which have the value 1 and those which have the value 0 respectively[16].

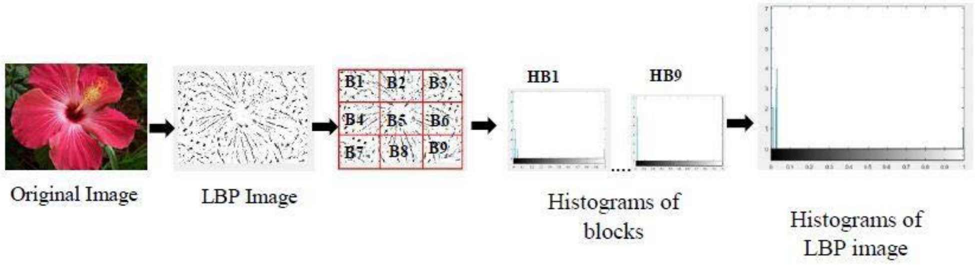


Figure 2: LBP coded image of a flower image

2.2 Distance Function

The concept of distance function has been developed to provide a formal description for measuring distance between two points in a vector space. In the section, we discuss about six distance function such as L2, chebyshev, cosine, correlation, city block and minkowski. These functions are used in our work to measure similarities between query image and the database images.

L2 Distance Function: The L2 distance between two points a and b is the length of the line segment between them.

If $a = (a_1, a_2, \ldots a_m)$ and $b = (b_1, b_2, \ldots b_m)$ are two points in Euclidian n-space then the distance d between a and b is given by

$$(a,b)=\sqrt{(a_1-b_1)^2+(a_2-b_2)^2+\cdots+(a_m-b_m)^2}$$

$$= \sqrt{\sum_{j=1}^{m}(a_j - b_j)^2} \tag{10}$$

where a and b is represented as Euclidian vectors which are starting from the origin of the space with their terminal points and ending at the two points.

$$\text{Euclidean norm of a is, } \|a\| = \sqrt{a.a} \;=\; \sqrt{{a_1}^2 + {a_2}^2 + \;...+{a_m}^2} \tag{11}$$

$$\text{And} \qquad\qquad \text{b is, } \|b\| = \sqrt{b.b} \;=\; \sqrt{{b_1}^2 + {b_2}^2 + \;...+{b_m}^2} \tag{12}$$

The L2 distance between a and b is

$$\|b - a\| = \sqrt{(b-a).(b-a)} \tag{13}$$

$$= \sqrt{\|a\|^2 + \|b\|^2 - 2a.b} \tag{14}$$

Chebyshev Distance Function: This distance function examines the entire magnitude of the difference between coordinates of a couple of objects. We use this function for measuring the similarity between the query flower image and the flower images into the data set.

The formula is, d_s=|x_{mk}−x_{nk}|I (15)

In Chebyshev distance, all 8 neighboring cells from the given point can be reached by one unit [17].

Cosine Distance function: We use Cosine Similarity for finding the similarity between the query flower image and the dataset flower images. It finds similarity between two nonzero vectors of an inner product space. It measures the cosine of the angle between the vectors.

The cosine of two non-zero vectors is derived from the Euclidean dot product formula [18] *ie.* $P.Q$=$\|P\|\ \|Q\|\ cos\theta$ (16)

Then the similarity is measured as,

$$Similarity, \qquad cos\theta = \frac{P.Q}{\|P\|\ \|Q\|} \tag{17}$$

$$= \frac{\sum_{i=1}^{n} P_i Q_i}{\sqrt{\sum_{i=1}^{n} {P_i}^2}\ \sqrt{\sum_{i=1}^{n} {Q_i}^2}} \tag{18}$$

Where, $P_i\, Q_i$ are the components of $P\ Q$ vectors.

Correlation: Correlation measures similarity rather than dissimilarity or distance. It is standardized angular separation by centering the coordinates to its mean value.Correlation distance function calculates both linear and nonlinear association between two or random vectors. Firstly, it calculates the distance correlation between two random vectors. Then, compare this value to the distance correlations of many shuffles of the data.

The formula is [19],

$$d_s = \frac{\sum_{i=1}^{n}(x_{pi} - \overline{x_p}).(x_{qi} - \overline{x_q})}{\left(\sum_{i=1}^{n}(x_{pi} - \overline{x_p})^2 . \sum_{i=1}^{n}(x_{qi} - \overline{x_q})^2\right)^{\frac{1}{2}}} \tag{19}$$

City block distance function: The City block distance is explained as if we consider two points in the xy-plane , the City block distance is instead measured as the distance in x plus and the distance in y. This is similar to the way if we move in a city, instead of going straight through we have to move around the buildings.

The City block distance between two points, p and q, with the dimensions n is determined as [20]:

$$City\ block\ distance\ , \quad CD = \sum_{i=1}^{n}|p_i - q_i| \quad , \quad CD \geq 0 \tag{20}$$

CD=0,r $identical\ points$ CD>0,$for\ Showing\ little\ similarity\ points$

The given figure displays an example of two points namely *p* and *q*. There are five values for each point *i.e. i =5*. The dotted lines are the distances (*p1-q1*), (*p2-q2*), (*p3-q3*), (*p4-q4*) and (*p5-q5*) which are go into in the above equation.

Results produced by city block distance are similar to the Euclidean distance. With this distance function in a single dimension, the effect of a large difference is reduced as the distances are not squared.

Minkowski Distance Function: The Minkowski distance is a metric in a normed vector space which can be considered as a generalization of both the Euclidean distance and the Manhattan distance.

The Minkowski distance of order m between two points [21]

$$U = (u_1, u_2, \dots, u_n)\ and\ V = (v_1, v_2 \dots v_n) \in \mathbb{R}^n \tag{21}$$

$$D(U,V) = \left(\sum |u_i - v_i|^m\right)^{\frac{1}{m}} \tag{22}$$

When, m<1, the distance between (0, 0) and (1,1) is $2^{1/m}$ >2

3 Experimental Results and Discussion

This section firstly discusses the image dataset construction and then the experiments to evaluate the proposed method using the dataset are carried out.

3.1 Image Dataset

To make flower image database, we download the pictures of flowers from Google. The dataset consists of 500 flower images of 20 species and there are 25 flower images of each specie. It is a challenging dataset as some flowers have similar types of texture description but those flowers are not of the same categories. Thus, it is difficult to separate the flower images. We actually arrange our flower dataset in this way so that this type of problems can arise and we try to introduce a method to solve those problems. Figure 2 represents a chunk of our dataset of different flower images.

Figure 2: Example flower images of dataset

3.2 Retrieval Results on Individual Feature

We calculate our results with a target for retrieving 20 similar images from our constructed dataset. For each feature and a single distance function, we observe the retrieval outcomes of our system on 5 types of flowers at least .Then, we take the average of retrieval accuracies corresponding to 5 types of flowers. The retrieval results for wavelet moments, Gabor wavelet and LBP feature and for several distance functions are shown in Table 1, Table 2, and Table 3 respectively. Indeed, we calculate retrieval accuracy for using wavelet moments, Gabor wavelet and LBP on the same set of distance functions.

In Table 1, we observe that the maximum result for the wavelet moments is 20% for Chebychev distance function (bold-faced). Also, we observe that the maximum result for the Gabor wavelet is 20% for city block and Chebychev distance function in Table 2. The maximum result for the LBP is 15% for city block distance function in Table 3. Note that there is a variation in retrieval results for the different distance functions in different features. However, a comparison among retrieval rates for wavelet moments, Gabor wavelet and LBP is represented in Figure 3. It is observable that the result for LBP feature is inferior compared to wavelet moments and Gabor wavelet. The maximum result for LBP feature is 15% for city block distance function and the correlation distance function. The city block and the cosine distance function are better for Gabor wavelet feature than the other distance functions. Using this distance function, we get 20% and 15% accurate result respectively for Gabor wavelet feature. But the Chebychev distance function is better for wavelet Moments. For this function, wavelet moments feature provides 20% accurate result

Table 1: Retrieval accuracy using wavelet moments feature

Distance function	Retrieval target	No. of retrieve image	Average Accuracy (%)
L2	20	2	10
City block	20	3	15
Minkowski	20	3	15
Chebychev	**20**	**4**	**20**
Cosine	20	2	10
Correlation	20	3	15

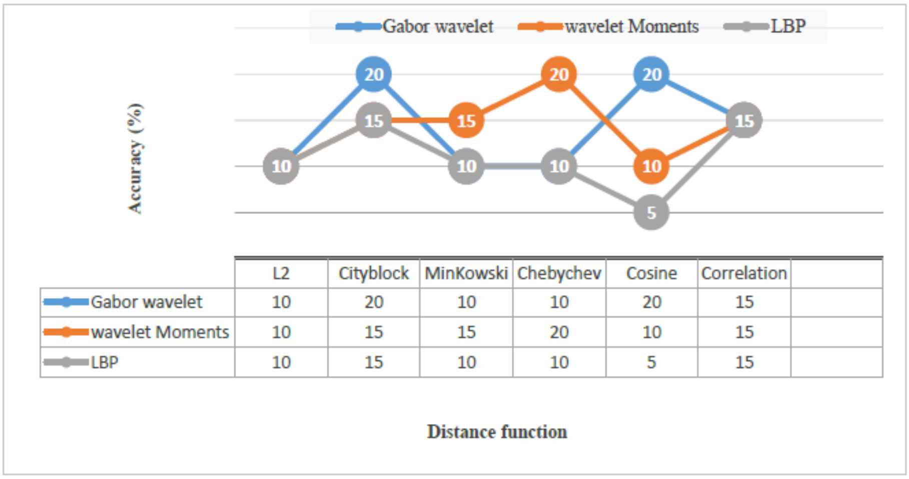
Table 2: Retrieval accuracy using Gabor wavelet feature

Distance function	Retrieval target	No. of retrieve images	Average Accuracy (%)
L2	20	2	10
City block	**20**	**4**	**20**
Minkowski	20	2	10

Chebychev	20	2	10
Cosine	**20**	**4**	**20**
Correlation	20	3	15

Table 3: Retrieval accuracy using LBP feature

Distance function	Retrieval target	No. of retrieve image	Average Accuracy (%)
L2	20	2	10
City block	**20**	**3**	**15**
Minkowski	20	2	10
Chebychev	20	2	10
Cosine	20	1	5
Correlation	20	3	15



	L2	Cityblock	MinKowski	Chebychev	Cosine	Correlation
Gabor wavelet	10	20	10	10	20	15
wavelet Moments	10	15	15	20	10	15
LBP	10	15	10	10	5	15

Figure 3: Comparison of retrieval rates for wavelet moments, Gabor wavelet and LBP, features

3.3 Retrieval Results on Combined Feature

To improve the retrieval outcome, the combination of wavelet moments, Gabor wavelet, and Local Binary Pattern (LBP) is considered. Table 4 shows the retrieval result on the concatenation of those features. The highest retrieval result of 35% is achieved when the cosine distance function is employed on the output of the integration of wavelet moments, Gabor wavelet, and Local Binary Pattern (LBP) feature vectors. It is worth to mention that the proposed system improves the retrieval accuracy by 15% when the wavelet moments, Gabor wavelet, and Local Binary Pattern (LBP) are combined rather than using them alone.

Table 4: Retrieval accuracy on the combination of wavelet moments, Gabor wavelet, and LBP

Distance Function	Retrieval Target	No. of Retrieve Image	Average Accuracy (%)
L2	20	4	20
City block	20	5	25
Minkowski	20	5	25
Chebychev	20	6	30
Cosine	**20**	**7**	**35**
Correlation	20	6	30

4 Conclusion

In this paper, we have calculated separately the retrieval accuracy for retrieving similar type of flower images for three descriptors such as Wavelet moments, Gabor wavelet and LBP. We observe that the result for wavelet moments and LBP feature are inferior compared to Gabor wavelet. We have also calculated retrieval outcomes for combining Wavelet moments, Gabor wavelet and LBP. The retrieval results gain the improvement for the combination of those features and better than the situation when the features are utilized alone. Furthermore, there is a variation in results for the different distance functions with the features. Some distance functions are giving higher results for some features only. The City block and correlation functions are good for all the features, whereas the minkowski and the chebychev function is good for only some specific features instead of all.

REFERENCES

[1] Nilsback M, Zisseman A. “A Visual Vocabulary for Flower Classification,” Institute of Electrical and Electronics Engineers (*IEEE) Computer Society Conference on Computer Vision and Pattern Recognition,*vol.2, pp.1447-1454. 2006..

[2] Nilsback M, Zisseman A. “Automated Flower Classification over a Large Number of Classes,” *Sixth Indian Conference on Computer Vision,Graphics&Image Processing*,pp.722-729, Dec.2008.

[3] M. Varma and D. Ray. Learning the discriminative power invariance trade-off. In Proc. International Conference on Computer Vision(ICCV), 2007.

[4] Ke, Xiao, Shaozi Li, and Xiaofen Chen. "Modified model in content-based flower image retrieval." *Intelligent Computing and Intelligent Systems (ICIS), 2010* Institute of Electrical and Electronics Engineers (*IEEE) International Conference on*. Vol. 3. IEEE, 2010.

[5] [5] Hsu, Tzu-Hsiang, Chang-Hsing Lee, and Ling-Hwei Chen. "An interactive flower image recognition system." *Multimedia Tools and Applications* 53.1 (2011): 53-73.

[6] Hong A, Chen G, Li J, Chi Z, Zhang D (2004) A flower image retrieval method based on ROI feature. Journal of Zhejiang University (Science) 5(7):764–772

[7] Zou, Jie, and George Nagy. "Evaluation of model-based interactive flower recognition." *null*. Institute of Electrical and Electronics Engineers (*IEEE)*, 2004

[8] Saitoh, Takeshi, Kimiya Aoki, and Toyohisa Kaneko. "Automatic recognition of blooming flowers." *null*. Institute of Electrical and Electronics Engineers (*IEEE)*, 2004.

[9] Hu, Xuelong, et al. "Flower image retrieval based on saliency map." *Computer, Consumer and Control (IS3C), 2014 International Symposium on*. Institute of Electrical and Electronics Engineers (*IEEE)*, 2014.

[10] Rafael C. Gonzalez, and E. Richard, “Digital Image Processing,” Prentice Hall Press, Second Edition, 2002.

[11] Ghazali, Kamarul Hawari, et al. "Feature extraction technique using discrete wavelet transform for image classification." *Research and Development, 2007. SCOReD 2007. 5th Student Conference on*. IEEE, 2007.

[12] Zhang, Dengsheng, et al. "Content-based image retrieval using Gabor texture features." Institute of Electrical and Electronics Engineers (*IEEE) Transactions Pattern Analysis and Machine Intelligence (PAMI)* (2000): 13-15.

[13] https://en.wikipedia.org/wiki/Local_binary_patterns

[14] Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture Classification with local binary patterns. Institute of Electrical and Electronics Engineers (*IEEE)* Trans. Pattern Anal. Mach. Intell. 24 (7), 971–987 (2002)

[15] Bulbul, Mohammad Farhad, Yunsheng Jiang, and Jinwen Ma. "Real-Time Human Action Recognition Using DMMs-Based LBP and EOH Features." *International Conference on Intelligent Computing*. Springer, Cham, 2015.

[16] T.Prathiba, G. Soniah Darathi, "An efficient content based image Retrieval using local tetra pattern," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 10, October 2013.

[17] https://lyfat.wordpress.com/2012/05/22/euclidean-vs-chebyshev-vs-manhattan-distance/

[18] https://en.wikipedia.org/wiki/Cosine_similarity

[19] http://people.revoledu.com/kardi/tutorial/Similarity/Correlation.html

[20] https://docs.tibco.com/pub/spotfire/7.0.0/doc/html/hc/hc_city_block_distance.htm

[21] https://en.wikipedia.org/wiki/Minkowski_distance

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

AIVP ADVANCES IN
IMAGE AND VIDEO PROCESSING

VOLUME 6, ISSUE 5
ISSN 2054-7412

Searching Human Action Recognition Accuracy from Depth Video Sequences Using HOG and PHOG Shape Features

Sadiya Tabussum, Javed Bin Al Amin, Mohammad Farhad Bulbul
Department of Mathematics,
Jessore University of Science and Technology, Jessore, Bangladesh
farhad@just.edu.bd

ABSTRACT

Research on human action recognition from depth video sequences are increasing day by day due to its vast application in automatic surveillance systems, entertainment environments, and healthcare systems etc. In our project, we improve human action recognition accuracy using shape features. We use Histogram of oriented gradients (HOG) and Pyramid Histogram of oriented gradients **(**PHOG) to extract shape features. The feature extraction algorithms are used to extract shape feature from dataset of different action videos. At first, depth motion maps (DMMs) are constructed from every action video. Then, the HOG and PHOG features are extracted from each DMMs. Using these features, actions are recognized by the l_2-regularized Collaborative Representation Classifier (l_2-CRC). In this paper, we evaluate our proposed method on MSR-Action 3D dataset. We divide this dataset into four action subsets such as AS1, AS2, AS3, and AS4 where each of them contains five actions. We compute the recognition accuracy of each action set using HOG and PHOG features respectively. Then, we take the comparison between the recognition accuracies of actions in every action set using HOG and PHOG features. Finally, we obtain the maximum recognition accuracy from most of the action subsets using PHOG feature. And the remaining subsets give poor results using HOG feature because of confusion between actions in those sets.

Keywords: Human action recognition, Depth motion maps, Histogram of oriented gradients, Pyramid Histogram of oriented gradients.

1 Introduction

Human action recognition is a significant area of computer vision research today. Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high dimensional data from the real world in order to produce symbolic information. The goal of human action recognition is to automatically analyze ongoing actions from an unknown video. The human action recognition is the process of detecting & labeling of all occurring action from an input video [1]. There are different types of actions based on difficulties such as gestures, human actions, interactions, and group activities [2]. Gestures are basic movements of a person's body portion, and are the nuclear components describing the meaningful motion of a person. "Spreading an arm" and "moving a leg" are good examples of gestures. Human actions are activities by single-person that may be a collection of more than one gestures prepared temporally, such as "walking", "waving", and "punching". Interactions are human activities that include two persons and/or objects. For example, "two-person handshaking" is an interaction between two humans and "a person stealing a travel bag in an airport" is a human-object interaction. Finally, the activities which are done by

DOI: 10.14738/aivp.65.5340
Publication Date: 4th November, 2018
URL: http://dx.doi.org/10.14738/aivp.65.5340

groups made of numerous persons and/or objects "A group of a player playing a game", "a group having a meeting," and "two groups fighting" are typical examples. In this research, the main focus is given to improve the recognition accuracy of human actions from video sequences.

Nowadays, more and more people record their daily activities using digital cameras, and this brings the enrichment of video content on the internet, and also causes the problems of categorizing the subsisting video, and sorting new videos according to the action classes present. Categorizing these videos is a time-consuming task if it is done manually, and recognizing certain actions is impossible to accomplish through manual effort. For these causes, the area of human action recognition has interested considerable attention [1].

Previously, researches were dependent on recognizing human action from image sequences taken by RGB cameras and the typical RGB input devices are color TV, video cameras, image scanners, and digital cameras [3]. The image from RGB camera is called RGB image in which every color pixel is made of red, green & blue color. Various constraints relating to RGB cameras are responsible to background clutter, camera motion, occlusions and illumination variations. So, it has been a tough and difficult task to precisely recognize human actions [4, 5]. However, with the development of cost-effective RGB depth (RGB-D) camera sensors, the results from action recognition have improved, and they have become a point of consideration for many researchers. The Microsoft Kinect (see Figure: 1) is an example of Kinect sensor [4, 5]. It includes a RGB camera, 3D depth sensor cameras, a tilt motor, multi-array microphone and LED light [6]. Depth sensors help lessen and ease the complications found in RGB images, such as background subtraction and light variations. Also, depth camera can be beneficial for the entire range of day-to-day work, even at night. So, it has been a big challenge to utilize these data, together or independently, to present human behavior and to improve the accuracy of action recognition [5]. Depth sensor camera provides RGB color image, depth image/map, skeleton, cardiovascular, muscular, nervous information extraction, facial and voice recognition, virtual therapy, patient information, x-ray, MRI, CT scans etc [7]. Depth map is a 3D image formed of gray pixels defined by 0~255 values. Various research works on human action recognition have been carried out based on depth maps and we will discuss about these works in literature review. An example of a depth map sequence is shown in Figure 2.

Depth images also enable us to view and assess human skeleton joints in a 3D coordinate system. These 3D skeleton joints provide additional information to examine for recognition of action, which in turn increases the accuracy of the human–computer interface. However, though some of researches based on the skeleton information show high recognition performance, they are not suitable in the case where skeleton information is not available [4].

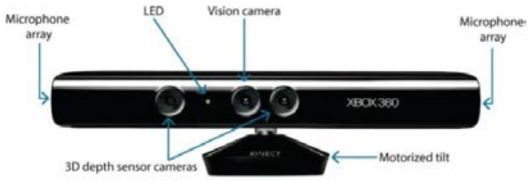
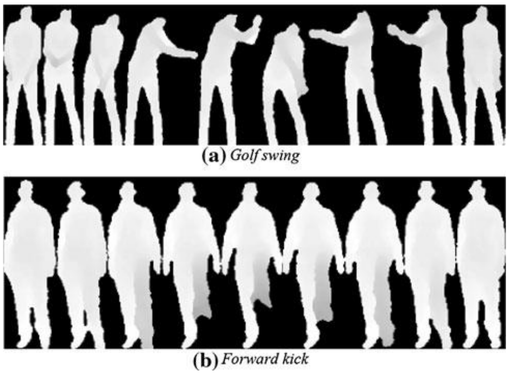


Figure 1: Kinect camera [8]



(a) Golf swing

(b) Forward kick

Figure 2: A depth map sequence for Golf swing & forward kick [9]

Figure 2: A depth map sequence for Golf swing & forward kick [9]

In this paper, Depth maps based action recognition system is proposed by representing an action with human shape in motion situation, representation and classification techniques. At first, all the video frames for each depth video are projected onto three orthogonal Cartesian planes so that the projected maps corresponding to three projection views (front, side, and top) to generated. For each projection sight, the addition of vital differences between consecutive projected maps forms Depth Motion Maps (DMMs) (i.e., DMM_f, DM , and DMM_t) [4]. DMMs is the combination of depth maps. From this DMMs, we extract shape features through HOG (Histograms of Oriented Gradients) [10] and PHOG (Pyramid Histograms of Oriented Gradients) [11] descriptors. Then the dimension of this features is reduced by statistical procedure named Principal Component analysis (PCA) [12] and human action is recognized by using l_2-CRC (l_2-regularized collaborative representation classifier) [4] algorithm. The proposed approaches are primarily evaluated with MSR-action 3D dataset [13] which is specifically designed for human action recognition.

2 Related Work

In this section, we discuss about the recent related work for human action recognition from depth map sequences.

In 2010, Li et al. [14] presented a method to recognize human actions from sequences of depth maps. They engaged an action graph which model the temporal dynamics of actions, and used a combination of 3D points to characterize postures. This approach contained some limitations are the loss of spatial context information between interest points and computational inefficiency. To improve recognition accuracy, In 2012 Vieira et al. [15] presented Space-Time Occupancy Patterns (STOP), a new optical demonstration for 3D action recognition from depth motion maps. In the same year Wang et al. [16] represented 3-D action sequences as 4-D shapes and proposed Random Occupancy Pattern (ROP), and sparse coding was used to further improve the toughness of the proposed approach. To improve recognition rates, Yang et al. [17], used Histogram of Oriented Gradients (HOG) features from DMMs and SVM to classify action. In the same year, Oreifej et al. [18] presented a new descriptor called Histogram of Oriented 4D Normals (HON4D) for activity recognition from videos. Luo et al., in 2014 [19], extracted collection of 3D features such that both the spatial and temporary features of the RGB sequences for each depth video by using Centre-Symmetric Motion Local Ternary Pattern (CS-Mltp). Lu et al. [20] proposed binary range-sample feature descriptor in depth. In 2015 Chen et al. [21], used texture feature local binary patterns (LBPs) to recognize action. In the same year Farhad et al. [22], used Depth Motion Maps (DMMs), Con-tourlet Transform (CT) [23] and Histogram of Oriented Gradients (HOGs) in order to distinguishing actions. In the next year Chen et al. [24], presented an effective local spatio temporal descriptor, the local binary patterns (LBP) [25] descriptor and used kernel-based extreme learning machine classifier. Chen et al. [26], proposed action recognition method by using a distance-weighted Tikhonov matrix an l2-regularized collaborative representation classifier (l2-CRC). In 2012, Yang and Tian et al. [27], used Naive-Bayes-Nearest-Neighbor (NBNN) [28] classifier for multi-class action classification from human skeleton. To get more accuracy, in the same year, Xia et al. [29] used histograms of 3D joint locations (HOJ3D) as a solid representation of poses for recognizing human action. In 2013 Luo et al. [30], submitted Dictionary Learning (DL) method and used the Temporal Pyramid Matching (TPM) which keep the temporal information so that they can recognize human action. Wang et al [31], In 2011, proposed a method for action recognition using Pyramid Histogram of Orientation Gradient (PHOG) shape features and They adopted two state-space models, i.e., Hidden Markov Model (HMM) [32] and Conditional Random Field (CRF) [33] to model the dynamic human movement.

The rest of this paper is organized as follows. The proposed approach is presented in Section 3. The experimental results are demonstrated in Section 4. Finally, Section 5 contains a brief conclusion of this work.

3 Our approach

In this section, the approach for recognizing action is discussed. The whole recognition approach can be divided into two phases: the training phase and the testing phase shown in Figure 3.

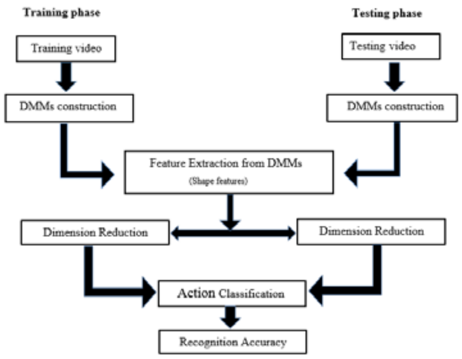


Figure 3: The proposed action recognition approach

During the training phase, DMMs (Depth motion maps) are constructed for each action, the shape feature is extracted from DMMs of each depth video obtained from the training video sequences. Histogram of oriented gradients (HOG) [10] and Pyramid Histogram of oriented gradients **(**PHOG) [11] descriptors are used to extract shape feature. Dimension of feature vector is reduced by Principal Component Analysis (PCA) [34].

During the testing phase, the same steps are followed to extract shape feature, build descriptors, reduce dimension as those done during the training phase. Then, l2-regularized Collaborative Representation Classifier (l2-CRC) [4] is adopted to train model and classify each testing sequence to the most probable action type.

3.1 DMMs construction

Depth Motion Maps (DMMs) was firstly proposed by Yang et al. [17]. In the feature extraction stage, by using generation techniques described in [26], DMMs are firstly constructed for each depth video sequence. Let F is the number of depth maps for each video sequence. The projection of each depth maps onto three orthogonal Cartesian planes provides three projected maps corresponding to the three projection views (front, side, and top). Let DM ,M_s and DMM_t are the three projected maps. The equation of each DMM is given bellow:

$$DMM_x = \sum_{n=1}^{F-1} \left| map_x^{n+1} - map_x^{n} \right| \tag{1}$$

Where, n = the frame index.

$x \in \{f, s, t\}$ denotes the projection view and map_x represents the projection map [4]. Figure 4 represents an example of DMMs for a tennis serve action video sequence

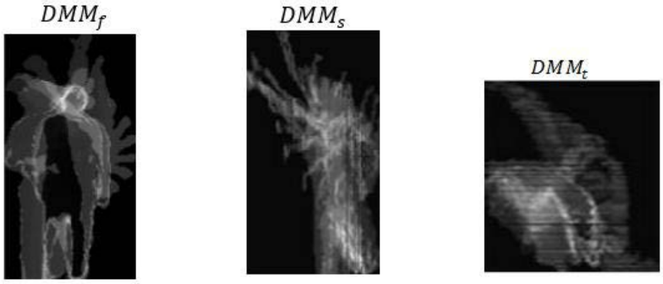


Figure 4: DMMs for a tennis serve action video sequence

3.2 Feature Extraction

We used Pyramid Histogram of oriented gradients (HOG) and Histogram of oriented gradients (PHOG) descriptor to extract shape features.

3.2.1 Histogram of oriented gradients (HOG)

The histogram of oriented gradients (HOG) is a hugely popular object descriptor. It has been shown to perform unpredictably well in human detection in still images as well as videos [35].The technique counts occurrences of gradient orientation in localized portion of an image-detection window [10].

The main goal of histogram of oriented gradients descriptor is to describe the local object shape feature within an image by the allocation of intensity gradients or edge directions. The gradient is a directional change in image intensity/color or measure of change in image function. The image is divided into small connected regions called cells, and for each cell a histogram of the centered horizontal and vertical gradient directions be computed for each pixel within the cell. Groups of neighboring cells are called blocks considered as spatial regions. Each cell is separated into angular bins according to the gradient orientation. Depending on the gradient magnitude is positive or negative, the histogram bins are equally distributed over 0 – 180 or 0 – 360. Finally, the edge orientations are quantized and the histograms of each block are normalized to compensate for brightness variation. Normalized group of histograms represents the block histogram. Then, the set of all normalized histograms obtained from all blocks represents the HOG descriptor [36]. The HOG on depth motion maps tennis serve is shown in figure 5.

Let, $(X,)$ is the image function. Where, X represents the horizontal direction and Y represents the vertical direction of cells.

Centered derivative:

$$f' = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h} \tag{2}$$

$$f_x = \frac{\partial f}{\partial x} = f(x+1) - f(x) = \text{Change of intensity in X} \quad \text{direction.} \tag{3}$$

$$f_y = \frac{\partial f}{\partial y} = f(y+1) - f(y) = \text{Change of intensity in Y direction.} \tag{4}$$

Gradient Vector:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right] = f_x\,\hat{\imath} + f_y\hat{\jmath} \tag{5}$$

Gradient Magnitude:

$$\|\nabla f\| = \sqrt{f_x^2 + f_y^2} \tag{6}$$

Gradient orientation:

$$\theta = tan^{-1}\left(\frac{f_y}{f_x}\right) \tag{7}$$

Normalization:

$$\hat{u} = \frac{\nabla f}{|\nabla f|} \quad [37] \tag{8}$$

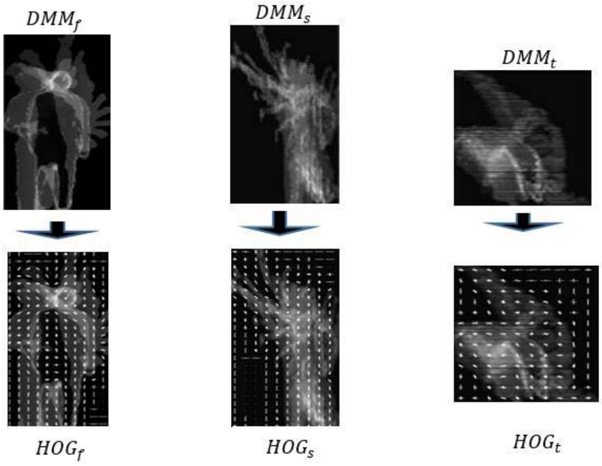
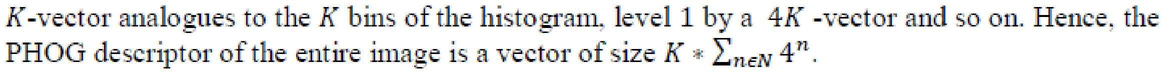


Figure 5: HOG on depth motion maps of tennis serve

3.2.2 Pyramid Histogram of oriented gradients (PHOG)

The pyramid of histogram of orientation gradients (PHOG) features are used to represent spatial shape descriptor. PHOG was proposed by Bosch et al. [38] and has been capably used in object classification. PHOG descriptor is used to represent an image by its local and global information of the shape. Pyramid histogram of gradients (PHOG) is an extension to HOG features. To find the PHOG feature image is partitioned into a sequence of small area of different resolutions by repeatedly doubling the division of area of interest at each level of the pyramid. Figure 6 shows the several pyramid levels. The small area of different levels called cells and a HOG feature vector are computed for every cell of different level. The combination of these HOG feature vectors represents the final PHOG descriptor. Hence, PHOG represents both edge direction and location. The histograms of oriented gradients (HOG) is mainly motivated the technique for extracting PHOG descriptor of Dalal and Triggs [39] to improve accuracy. Along both directions of 2D axis the image at level n is split into $2n$ cells to build the pyramid. Therefore, level 0 is described by a



K-vector analogues to the K bins of the histogram, level 1 by a $4K$ -vector and so on. Hence, the PHOG descriptor of the entire image is a vector of size $K * \sum_{n\epsilon N} 4^n$.

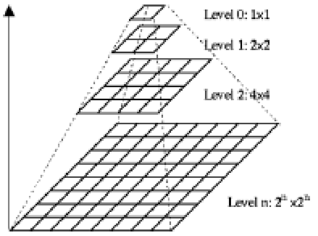


Figure 6: The several pyramid levels [47]

For example, for levels up to N = 3 and K = 9 bins the PHOG descriptor dimension will be $(9 * \Sigma 4n)$ 3 $n=0 = 765$. For any application, it is necessary to limit the number of levels of the pyramid to N = 3 to prevent over fitting [40].

3.3 Action classification with $\boldsymbol{l_2}$-regularized Collaborative Representation Classifier ($\boldsymbol{l_2}$- CRC)

$\boldsymbol{l_2}$-CRC is widely used classifier to recognize human action [9, 22]. To explain $\boldsymbol{l_2}$-CRC in details, with C classes let us consider a data set. An over-complete dictionary $S = [S_1, S_2, \dots \dots \dots, S_C] = [s_1, s_2, \dots \dots \dots, s_n] \in \mathbb{R}^{d\times n}$ can be obtained by arranging the training samples column wise.

Where,

$\boldsymbol{S}_j \in \mathbb{R}^{d\times}, (j = 1,2, \dots \dots \dots, C) =$ Subset of the training samples related to class

$\boldsymbol{s}_i \in \mathbb{R}, (i = 1,2, \dots \dots \dots, n) =$ Single training sample

d = dimension of training samples

n = total number of training samples from all classes

Using the matrix $\boldsymbol{S}$, let $\boldsymbol{X} \in \mathbb{R}^d$ can be expressed as any unknown sample. Which can be formulated as,

$$x = \boldsymbol{S}\beta, \tag{9}$$

Where, $\beta = [\beta 1, \beta 2, \dots \dots \dots, \beta j\,]$ is $n \times 1$ vector of coefficients analogous to all the training samples and $\beta j\ (j = 1,2, \dots \dots \dots, C)$ is the subset of the coefficients related with the training samples from the class j.

Effectively, one can not directly solve Equation (9) since it is naturally under-determined. Then the solution is obtained by solving the following norm minimization problem.

$$\hat{\beta} = \arg\underset{\beta}{min}\{\|\boldsymbol{x} - \boldsymbol{S}\beta\|_2^2 + \alpha\|\boldsymbol{M}\beta\|_2^2\}, \tag{10}$$

Where, $\alpha =$ the regularization parameter

$\boldsymbol{M} =$ The Tikhonov regularization matrix [41]

By applying the approach described in [42-44], the term related with $\boldsymbol{M}$ approve the assessment of preceding knowledge of the solution. Where, the weight of the training samples which are extremely different from a test sample are less than the training samples that are extremely similar. Particularly, the form of the matrix $\boldsymbol{M} \in \mathbb{R}^{d\times n}$ is represented as follows:

$$M = \begin{pmatrix} \|x - s_1\|_2 & 0 & \dots & 0 \\ 0 & \|x - s_2\|_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \|x - s_n\|_2 \end{pmatrix} \tag{11}$$

According to [45] the coefficient vector β is calculated as follows,

$$\hat{\beta} = (\boldsymbol{S}^T\boldsymbol{S} + \alpha\boldsymbol{M}^T\boldsymbol{M})^{-1}\,\boldsymbol{S}^T\boldsymbol{V} \tag{12}$$

^

$\hat{\beta}$ can be partitioned into C subsets $\hat{\beta} = [\hat{\beta}_1; \hat{\beta}_2; \dots \dots; \hat{\beta}_C]$ with $\hat{\beta}_j (j = 1,2, \dots \dots C)$

By using the class labels of all the training samples. After portioning β the class label of the unknown sample $\boldsymbol{x}$ is then calculated as follows:

$$class(\mathbf{x}) = \underset{j\in\{1,2,\dots,C\}}{\arg min}\{e_j\} \tag{13}$$

Where, $e_j = \|\boldsymbol{x} - \boldsymbol{S}_j\ \hat{\beta}\|_2$ [4].

4 Experimental Results and Discussion

In this section, we first evaluate our proposed method on publicly available MSR- Action 3D dataset then compare between the results obtained by using HOG and PHOG descriptor

4.1 MSR-Action 3D Dataset & Setup

MSR-Action 3D dataset is an action dataset of depth map sequences recorded by a depth camera. This dataset contains 20 action types performed by 10 different subjects and every subject performs each action 2 or 3 times. There are 557 depth map sequences in total. The resolution of each map is 320x240. The data was a depth sensor similar to the Kinect device. The 20 actions of this datasets are: “High arm wave”, “Horizontal arm wave”, “Hammer”, “Hand catch”, “Forward punch”, “High throw”, “Draw x”, “Draw tick”, “Draw circle”, “Hand clap”, “Two hand wave”, “Side-boxing”, “Bend”, “Forward kick”, “Side kick”, “Jogging”, “Tennis swing”, “Tennis serve”, “Golf swing”, “Pickup & throw” [46].

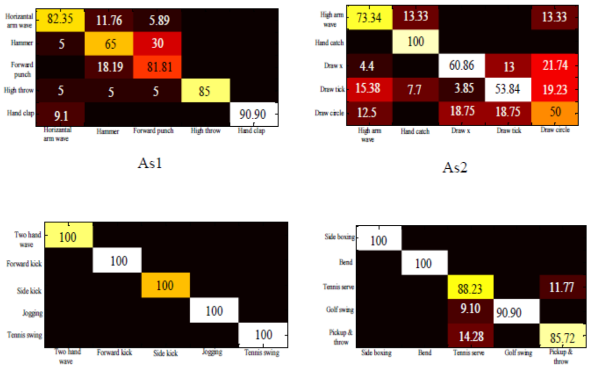
Table 1: Four action subsets of MSR-Action 3D dataset

Action Label	Action set 1 (AS1)	Action Label	Action set 2 (AS2)	Action Label	Action set 3 (AS3)	Action Label	Action set 4 (AS4)
2	Horizontal arm wave	1	High arm wave	11	Two hand wave	12	Side boxing
3	Hammer	4	Hand catch	14	Forward kick	13	Bend
5	Forward punch	7	Draw x	15	Side kick	18	Tennis serve
6	High throw	8	Draw tick	16	Jogging	19	Golf swing
10	Hand clap	9	Draw circle	17	Tennis swing	20	Pickup & throw

This dataset is very challenging dataset. To find out recognition accuracy using these 20 actions together, we have to use a computer with high configuration. But we don’t have such computer. So, we divide these 20 actions into four action subsets, those are AS1, AS2, AS3 and AS4 showed in Table 1. Each of them contains 5 actions. Two different test cases were performed on each action subset. For test one, 1/3 samples of each subset are used for training and the remaining samples for test; For test two, 2/3 samples of each subset are used for training and the remaining samples for test.

4.2 Action Classification result

To compute accuracy we use DMMs of size 320x240 to extract HOG & PHOG feature vectors. Dimension of these HOG & PHOG feature vectors are 216 & 2104 respectively. These dimensions are reduced by PCA. So, after dimensionality reduction, the new dimension of HOG & PHOG is 15 & 11. Then final feature vectors are fed into $\boldsymbol{l}_{\mathbf{2}}$-CRC to recognize human action and the key parameter α is a set as α=0.0001 in $\boldsymbol{l}_{\mathbf{2}}$-CRC. We find out our action classification accuracy of sets AS1, AS2, AS3 and AS4 of MSR-Action 3D dataset by using HOG and PHOG shape features and then classification accuracies of the each action set is compared obtained by using HOG and PHOG. We obtained our result using confusion matrix on the test data. Figure (7,8) represents the confusion matrix of action sets for test one and test two using HOG feature descriptor and Figure (9,10) represents the confusion matrix of action sets for test one and test two using PHOG feature descriptor.



As1

As2

As3

As4

Figure 7: The confusion matrix of action sets for test one using HOG feature descriptor

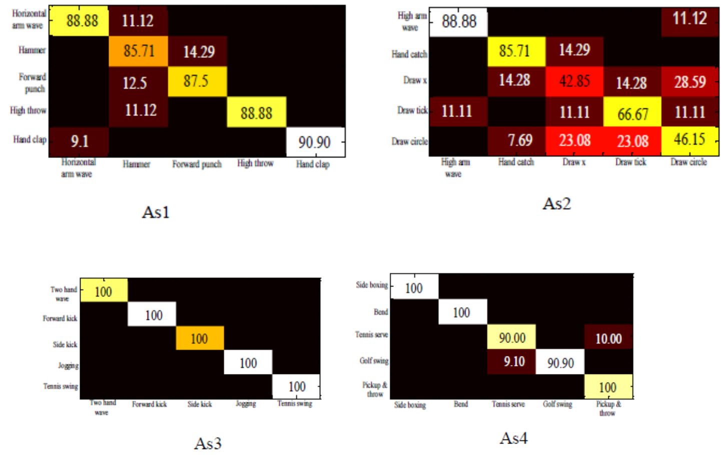


Figure 8: The confusion matrix of action sets for test two using HOG feature descriptor

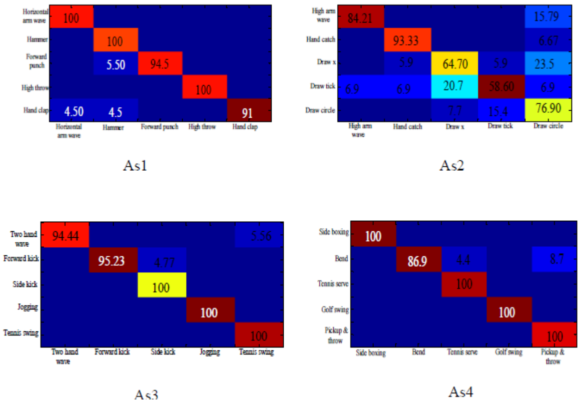


Figure 9: The confusion matrix of action sets for test one using PHOG feature descriptor

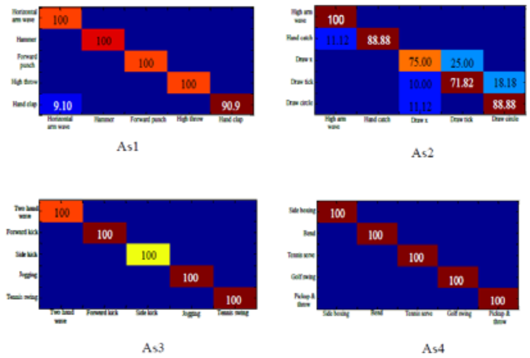


Figure 10: The confusion matrix of action sets for test two using PHOG feature descriptor

We compare between the recognition accuracies obtained by using HOG & PHOG descriptor to find out which one is the best. Figure 11-18 represent the comparison between the accuracies obtained by HOG and PHOG shape feature descriptor.

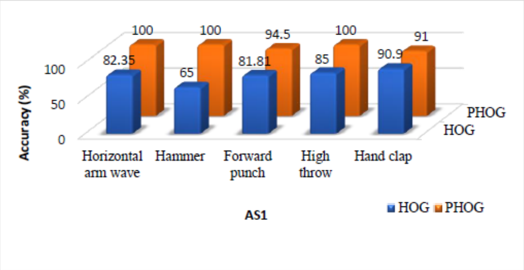


Figure 11: Comparison graph of actions in AS1 for test one using HOG and PHOG

In this case, PHOG is the best feature descriptor for AS1 test one comparing with HOG. Three actions in AS1 gives maximum accuracy for test one using PHOG feature. But the remaining two actions give poor result, because the action Forward punch is confused with Hammer and Hand clap is confused with Horizontal arm wave and Hammer.

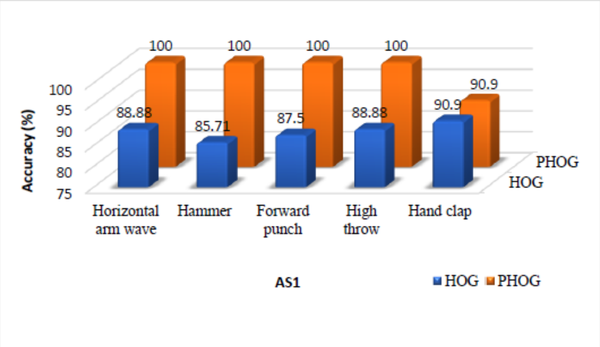


Figure 12: Comparison graph of actions in AS1 for test two using HOG and PHOG feature

In this case, PHOG is the best feature descriptor for AS1 test two comparing with HOG. Four actions in AS1 gives maximum accuracy for test two using PHOG feature. But, the remaining action Hand clap gives poor result, because this action is confused with Horizontal arm wave

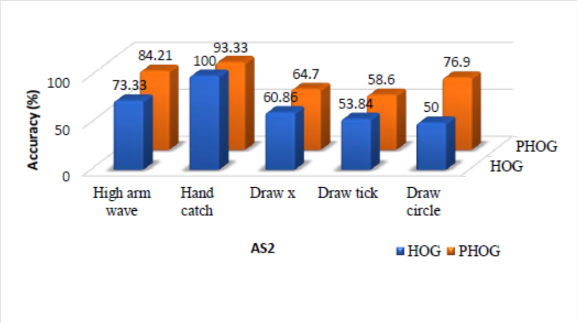


Figure 13: Comparison graph of actions in AS2 for test one using HOG and PHOG

In this case, Although HOG is the best feature descriptor for AS2 test one comparing with PHOG, only one action Hand catch in AS2 gives maximum accuracy for test one. And the remaining four actions gives poor result, because High arm wave is confused with Draw circle, Hand catch and Draw x are confused with each other, Draw x and Draw tick are confused with each other, Draw x is confused with Draw circle, and Draw tick is confused with High arm wave

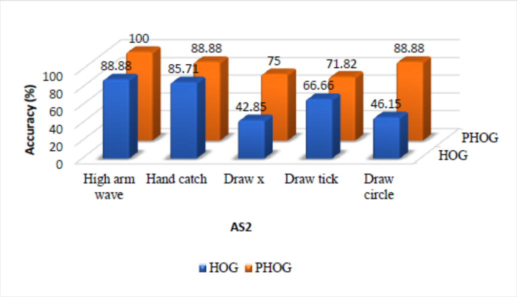


Figure 14: Comparison graph of actions in AS2 for test two using HOG and PHOG feature

In this case, Although PHOG is the best feature descriptor for AS2 test two comparing with HOG, only one action High arm wave in AS2 gives maximum accuracy for test two. And the remaining four actions gives poor result, the action Hand catch is confused with High arm wave, Draw x and Draw tick are confused with each other, Draw tick is confused with Draw circle, and Draw circle is confused with Draw x.

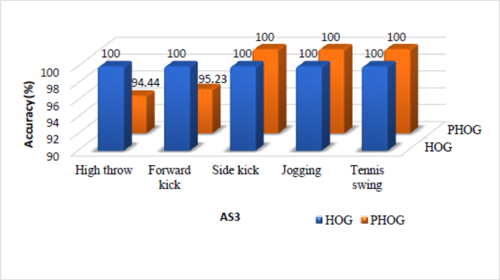


Figure 15: Comparison graph of actions in AS3 for test one using HOG and PHOG feature

In this case, HOG is the best feature descriptor for AS3 test one comparing with PHOG, because all actions in AS3 gives maximum accuracy. But, only three actions give maximum accuracy using PHOG feature

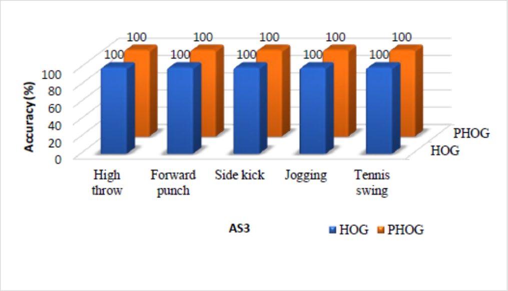


Figure 16: Comparison graph of actions in AS3 for test two using HOG and PHOG feature

In this case, all actions in AS3 gives maximum accuracy for test two using HOG and PHOG feature. So, HOG & PHOG both are the best feature descriptors for AS3 test two comparing each other.

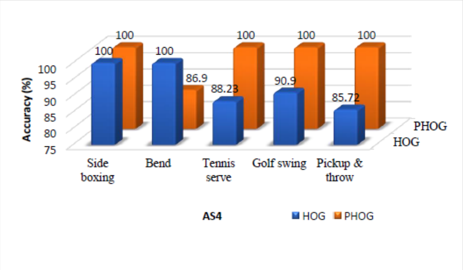


Figure 17: Comparison graph of actions in AS4 for test one using HOG and PHOG feature

In this case, PHOG is the best feature descriptor for AS4 test one comparing with HOG. Four actions in AS4 gives maximum accuracy for test one using PHOG feature. But the remaining action gives poor result, because this action Bend is confused with the action Tennis serve.

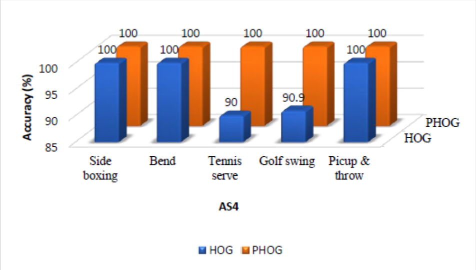


Figure 18: Comparison graph of actions in AS4 for test two using HOG and PHOG feature

In this case, PHOG is the best feature descriptor for AS4 test two comparing with HOG, because all actions in AS4 gives maximum accuracy for test two using PHOG.

5 Conclusion

In this paper, depth map based human action recognition method using Histogram of Oriented Gradients (HOG) and Pyramid Histogram of Oriented Gradients (PHOG) as feature descriptors is introduced. We use HOG because, it gives local feature but we also used PHOG so that we can get both the global and local information. Our experiment has been carried out on MSR-action 3D dataset that both the HOG and PHOG features are extracted from DMMs of every action of four action set AS1, AS2, AS3, and AS4 of MSR-action 3D dataset. The implementation has been done exclusively using MATLAB version R2014a on Lenovo with 8 GB RAM, CORE i5 processor, and win10 system. Then the experimental results obtained by using HOG feature and those of using PHOG feature for AS1, AS2, AS3, and AS4 are compared. Using HOG feature the classification accuracy of AS2 for test one is 100% for one action, AS3 for test one and test two is 100% for all actions. And using PHOG feature the classification accuracy of AS1 for test one is 100% for three actions and for test two is 100% for four actions, AS2 for test two is 100% for one action, AS3 for test two is 100% for all actions and AS4 for test one is 100% for four actions and test two is100% for all actions. But using PHOG, maximum accuracy of AS2 for test one is 93.33% for one action where HOG gives 100% accuracy for one action and AS3 for test one is 100% for three actions where HOG gives 100% accuracy for all actions. So, PHOG with HOG feature, AS2 for test one and AS3 for test one gives the worst result, because in AS2 the action High arm wave is confused with Draw circle, Hand catch is confused with Draw circle, Draw x is confused with Hand catch, Draw tick and Draw circle, Draw tick is confused with High wave, Hand catch and Draw x; and in AS3 the action Two hand wave is confused with Tennis swing, and forward kick is confused with side kick. This confusion occurs because, the local features of subjects can't be detected properly. Overall, PHOG gives the best result to recognize human action.

REFERENCES

[1] Umakanthan, Sabanadesan. *Human action recognition from video sequences*. Diss. Queensland University of Technology, 2016.

[2] Aggarwal, Jake K., and Michael S. Ryoo. "Human activity analysis: A review." *ACM Computing Surveys (CSUR)* 43.3 (2011): 16.

[3] https://en.wikipedia.org/wiki/RGB_color_model

[4] Bulbul, Mohammad Farhad, Yunsheng Jiang, and Jinwen Ma. "Real-Time Human Action Recognition Using DMMs-Based LBP and EOH Features." *International Conference on Intelligent Computing*. Springer, Cham, 2015.

[5] Farooq, Adnan, and Chee Sun Won. "A survey of human action recognition approaches that use an RGB-D sensor." *IEIE Transactions on Smart Processing & Computing* 4.4 (2015): 281-290.

[6] Chen, Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. "A real-time human action recognition system using depth and inertial sensor fusion." *IEEE Sensors Journal* 16.3 (2016): 773-781.

[7] https://www.hongkiat.com/blog/innovative-uses-kinect/

[8] https://www.researchgate.net/figure/Architecture-of-Microsoft-Kinect-sensor_fig10_282477283

[9] Chen, Chen, Kui Liu, and Nasser Kehtarnavaz. "Real-time human action recognition based on depth motion maps." *Journal of real-time image processing* 12.1 (2016): 155-163.

[10] https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients

[11] http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.htm

[12] ftp://statgen.ncsu.edu/pub/thorne/molevoclass/AtchleyOct19.pdf

[13] http://users.eecs.northwestern.edu/~jwa368/my_data.html

[14] Li, Wanqing, Zhengyou Zhang, and Zicheng Liu. "Action recognition based on a bag of 3d points." *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 2010.

[15] Vieira, Antonio W., *et al*. "Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences." *Iberoamerican congress on pattern recognition*. Springer, Berlin, Heidelberg, 2012.

[16] Wang, Jiang, *et al*. "Robust 3d action recognition with random occupancy patterns." *Computer vision–ECCV 2012*. Springer, Berlin, Heidelberg, 2012. 872-885.

[17] Yang, Xiaodong, Chenyang Zhang, and YingLi Tian. "Recognizing actions using depth motion maps-based histograms of oriented gradients." *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 2012.

[18] Oreifej, Omar, and Zicheng Liu. "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013.

[19] Luo, Jiajia, Wei Wang, and Hairong Qi. "Spatio-temporal feature extraction and representation for RGB-D human action recognition." *Pattern Recognition Letters* 50 (2014): 139-148.

[20] Lu, Cewu, Jiaya Jia, and Chi-Keung Tang. "Range-sample depth feature for action recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

[21] Chen, Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. "Action recognition from depth sequences using depth motion maps-based local binary patterns." *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*. IEEE, 2015.

[22] Bulbul, Mohammad Farhad, Yunsheng Jiang, and Jinwen Ma. "Human action recognition based on DMMs, HOGs and Contourlet transform." *Multimedia Big Data (BigMM), 2015 IEEE International Conference on*. IEEE, 2015.

[23] https://en.wikipedia.org/wiki/Contourlet

[24] Chen, C., Liu, M., Zhang, B., Han, J., Jiang, J., & Liu, H. (2016, July). 3D Action Recognition Using Multi-Temporal Depth Motion Maps and Fisher Vector. In *IJCAI* (pp. 3331-3337).

[25] https://en.wikipedia.org/wiki/Local_binary_patterns

[26] Chen, Chen, Kui Liu, and Nasser Kehtarnavaz. "Real-time human action recognition based on depth motion maps." *Journal of real-time image processing* 12.1 (2016): 155-163.

[27] Yang, Xiaodong, and Ying Li Tian. "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor." *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on*. IEEE, 2012.

[28] Timofte, Radu, Tinne Tuytelaars, and Luc Van Gool. "Naive bayes image classification: beyond nearest neighbors." *Asian Conference on Computer Vision*. Springer, Berlin, Heidelberg, 2012.

[29] Xia, Lu, Chia-Chih Chen, and Jake K. Aggarwal. "View invariant human action recognition using histograms of 3d joints." *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on*. IEEE, 2012.

[30] Luo, Jiajia, Wei Wang, and Hairong Qi. "Group sparsity and geometry constrained dictionary learning for action recognition from depth maps." *Proceedings of the IEEE international conference on computer vision*. 2013.

[31] Wang, Jin, *et al.* "Human action recognition based on pyramid histogram of oriented gradients." *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*. IEEE, 2011.

[32] https://en.wikipedia.org/wiki/Hidden_Markov_model

[33] https://en.wikipedia.org/wiki/Conditional_random_field

[34] https://en.wikipedia.org/wiki/Principal_component_analysis

[35] https://www.quora.com/What-is-a-histogram-of-gradient-directions-in-computer-vision

[36] https://scc.ustc.edu.cn/zlsc/tc4600/intel/2017.0.098/ipp/common/ipp_manual/GUID-83B0EE35-E5EF-4C23-96A4-F0918DFA826B.htm

[37] https://www.youtube.com/user/Udacity

[38] Bosch, Anna, Andrew Zisserman, and Xavier Munoz. "Representing shape with a spatial pyramid kernel." *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 2007.

[39] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005.

[40] http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.htm

[41] Tikhonov, A., Arsenin, V.: Solutions of ill-posed problems. Math. Comput. 32(144), 1320–1322 (1978)

[42] Chen, C., Tramel, E.W., Fowler, J.E.: Compressed-sensing recovery of images and video using multi-hypothesis predictions. In: Proceedings of the 45th Asilomar Conference on signals, Systems, and Computers, pp. 1193–1198 (2011)

[43] Chen, C., Li, W., Tramel, E.W., Fowler, J.E.: Reconstruction of hyperspectral imagery from random projections using multi-hypothesis prediction. IEEE Trans. Geosci. Remote Sens. 52(1), 365–374 (2014)

[44] Chen, C., Fowler, J.E.: Single-image Super-resolution Using Multi-hypothesis Prediction. In: Proceedings of the 46th Asilomar Conference on Signals, Systems, and Computers, pp. 608–612 (2012)

[45] Golub, G., Hansen, P.C., O'Leary, D.: Tikhonov-regularization and total least squares SIAM J. Matrix Anal. Appl. 21(1), 185–194 (1999)

[46] http://users.eecs.northwestern.edu/~jwa368/my_data.html

[47] http://fourier.eng.hmc.edu/e161/lectures/canny/node2.html

AIVP ADVANCES IN IMAGE AND VIDEO PROCESSING

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

VOLUME 6, ISSUE 5
ISSN 2054-7412

Information Security in Civil Aviation

G.D.Zhangissina, Karimov T.A., Kim P.A., Kim M.V., Kazakhstan, Almaty
Vice-Prezident of International Academy of Informatization,
Kazakhstan, Almaty
gul_zhd@mail.ru

ABSTRACT

In this paper is considered Information Security in Civil Aviation. This is the more important topic. Because the Security of country dependens from Information Security in Civil Aviation.

Key words: information security, civil aviation, information, security, protection, transport, information, system, antivirus, range, company.

There are many definitions of information security (IB), for example:

- Information security is the process of ensuring the confidentiality, integrity and availability of information.

- information security - all aspects related to the definition, achievement and maintenance of confidentiality, integrity, availability, accountability, authenticity and reliability of information or means of its processing.

- information security is determined by the absence of unacceptable risk associated with information leakage through technical channels, unauthorized and unintended effects on data and (or) on other resources of an automated information system used in an automated system.

All definitions mean about the same thing: for example, you have a customer base, you want to be sure that only you have access to it, and at any time the information will be correct there. For example, the names and phone numbers of customers will be the same as what you entered there, and no one changed them. For this and need to protect the database.

The relevance of information security in the informatization of society. The relevance of information security is due to the ever-increasing informatization of society. The scope of application of computer technology is constantly increasing (telephones, finance, technological process management at enterprises, etc.), the amount of data processed in information systems is also increasing, and geographically distributed computing systems are also becoming increasingly common.

The more information is concentrated in the information system, the more willing to get it, and the more complex the system itself - the more potential vulnerabilities it has. But to abandon the use of information systems for solving various tasks is no longer possible, and it remains either to put up with the possibility of leaking important data, or to protect them.

Moreover, it is necessary to protect the information values accordingly, for example, antivirus, firewall should be installed on a personal computer and vigilance should be exercised when surfing the Internet (especially important when using online banking), and to ensure data protection in a company, you may need a whole range of measures and serious technical tools. protection.

DOI: 10.14738/aivp.65.5582
Publication Date: 4th October 2018
URL: http://dx.doi.org/10.14738/aivp.65.5582

Information security in civil aviation.

If we consider the relevance of information security for civil aviation, it is worth starting with the definition of information that requires protection. This may be information, the value of which we define ourselves, or information that has legally determined value.

1. Protection of information of value to the company:

- commercial secrets - a wide range of information; companies themselves refer certain data to commercial secrets, various financial information, marketing plans, customer bases, contact information of foreign partners, etc.

- building plans are information that may be physically damaged by the leakage; the same terrorist attack is preceded by gathering information, finding out the building plans, how cameras are placed in them, possibly a schedule and a plan for going around the premises.

- placement of cameras. Information to be protected by laws.

How to ensure information security in airlines?

Information security in the field of civil aviation is achieved by the timely adoption of a number of measures and certain means of protection in airlines, airports and in companies-counterparties.

1. Organizational measures. First of all, the protection of information should begin with a person, with an employee who will assume the responsibility to ensure information security in the company. If the company is very large, then one employee can not do, and it is better to create a department of at least 3 people, one manager and two engineers. In any case, the employee or the head of the department must have a good understanding of information security issues, the level of competence of the employee will necessarily affect the financial costs of information security. When an employee is found, after studying the information processes in the company, he must develop appropriate organizational and administrative documents, which are signed by the main person in the company.

2. Physical measures - restriction of physical access to the protected information.

3. Technical measures - various means of protecting against unauthorized access, protection of information from leakage through technical communication channels, cryptographic means of protecting information, systems for protecting against DDOS attacks.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM

AIVP ADVANCES IN
IMAGE AND VIDEO PROCESSING

VOLUME 6, ISSUE 5
ISSN 2054-7412

Performance Analysis of FPGA Based MAC Unit using DBTNS Multiplier & TRNS Adder for Signal Processing Algorithm

[1]Aniruddha Ghosh, [2]Amitabha Sinha
[1]Calcutta Institute Of Technology, Uluberia, Howrah, W.B, India;
[2]Birbhum Institute of Engineering & Technology, Suri, Birbhum, W.B, India;
g_aniruddha2003@yahoo.co.in; becas_amits83@hotmail.com;

ABSTRACT

Digital signal processing (DSP) algorithms are actually nothing but sum of product. So, they are computationally intensive. All the mathematical tasks related to DSP algorithm are based on multiplication and addition. So, the implementations of DSP algorithms-based applications extensively require multiplier and adder. Due to extensive use of multiplication and addition operation, speed up cannot be achieved. Designing the high-performance adder and multiplier are primary objective for implementing high performance signal processing applications. This hindrance can be removed by Multiply-Accumulate Unit (MAC). The special feature of a MAC unit is its ability to perform single cycle multiplication and addition operation.

The performance of MAC Unit can be improved by using non-binary number system. Ternary value logic (TVL) has the ability to offer several advantages over conventional binary number system like reduce chip area, reduce overall delay. TVL can switch between three levels, such levels denoted by 0, 1, 2. As Residue Number Systems (RNS) can perform "carry free" arithmetic operations, high performance adder can be implemented using RNS in TVL domain (TRNS). Partial product free multiplication can be implemented by using Double Base Number System in TVL domain (DBTNS). Double Base Ternary Number System (DBTNS) multiplier can perform better than conventional TVL multiplier. A MAC unit is used to perform the multiplication and accumulator operations together to avoid unnecessary overhead on the processor in terms of processing time and the on-chip memory requirements.

Keeping in view of these issues, a new architecture is proposed for implementing high performance MAC unit for DSP applications. In this paper, a new approach of designing MAC unit is instigated using DBTNS multiplier and TRNS adder. A major bottleneck of implementing this architecture is the complexity involved in converting TVL to DBTNS in initial stage and converting TRNS to TVL in final stage. The performance of Ternary Residue Number Systems (TRNS) system depends on selection of moduli because selection of moduli is not properly maintained then it will affect system speed, dynamic range and hardware complexity. Proposed MAC unit is mapped on field programmable gate array (FPGA) for analysis its performance.

Keywords: Ternary Logic Value (TVL), Trit, Ternary Resdue Number Systems (TRNS), Double Base Ternary Number System (DBTNS), DBTNS Multiplier, TRNS Adder, Multiply-Accumulate Unit (MAC), FPGA, DSP Algorithms.

1 Introduction

The numeral systems which support three level of switching is termed as Ternary Value Logic (TVL) whose base is 3 and each ternary digit is termed as trit[1]. Volume of Information stored in a trit is

DOI: 10.14738/aivp.65.5394
Publication Date: 4th October 2018
URL: http://dx.doi.org/10.14738/aivp.65.5394

$\log_2 3$. In Ternary Value Logic (TVL), 0, 1, and 2 are used to represent all numbers [2]. Almost all the Digital Signal Processing applications are mainly nothing but sum of product. In the current scenario, high speed processors having dedicated hardware is one of the main concerns. The enhancement can be made by designing high speed multiplications and additions unit [3]. Multiply-Accumulate (MAC) unit has the capabilities to handle the high-performance digital processing system for Digital Signal Processing (DSP) algorithms [4][5][6]. Significant building blocks of a MAC unit are multiplier, adder and accumulator [5]. In a MAC unit, initially, input data are multiplied then they are added with previously stored accumulator data. Due to extensive use of portable electronic systems like laptop, calculator, mobile etc., and the low power devices have become very popular in today's world [7]. Main aim of VLSI design is to implement low power and high-throughput circuitry design [8][9]. So, implementation of fast and efficient MAC unit is the key objective for real-time signal processing system [10]. This speed can be achieved by enhancing the speed of basic modules of MAC unit by making multiplier unit and adder unit fast. Non-weighted and non-binary number system can help to implement such fast unit [6]. The Residue Number System (RNS) [11] is a non-weighted number system. RNS breaks a large number into the set of smaller number depending on moduli set [12]. Moduli set must be relatively prime to achieve maximum dynamic range [13]. It is shown in various studies that RNS can able to handle fault tolerant, detect and correct the fault [14]. In recent days Residue Number Systems (RNS) [11][15] is attractive owing their competencies of performing carry free addition. So, high performance adder can be implemented using Ternary Residue Number System (TRNS) i.e. RNS using TVL. This enhancement in speed is achieved due to concurrent operations on the moduli. There is another non-weighted number system present, namely, Double Base Number System (DBNS) [16] which can able to perform partial product free multiplication. So, Double Base Ternary Number System (DBTNS) multiplier can help to reduce the complexity of multiplication in compare with the conventional Ternary multiplier. But major bottleneck is the extraction of indices ([i, j] pair) [16][17] when converting ternary number to double base number. For implementing DBTNS conversion, LUT based approach have been adopted. But when dynamic range increases, LUT based approach become incapable to break the complexity as the LUT size increase exponentially. Partial product free multiplication can be performed by DBTNS multiplier so high speed multiply accumulate (MAC) [4][5] units can be implemented using DBTNS multiplier. Keeping these issues in view, this paper presents a new architecture for efficient implementation of MAC units exploiting the potentials of Ternary Residue Number System (TRNS) Adder and Double Base Ternary Number System (DBTNS) Multiplier using Ternary Value Logic (TVL). Performance analysis of a MAC unit using such a scheme clearly indicates the novelty of the architecture. The architecture was implemented and validated on Xilinx Virtex FPGA [16] [17] [18].

2 Review of TVL and RNS

2.1 Review Of Ternary Value Logic (TVL)

In binary system, logic levels are restricted in two state either 0 or 1. So less amount of information is carried by binary logic. To overcome this restriction, the alternative can be suggested as multivalued logic system [2]. But one cannot use unlimited logic level as limitation depends on use of technology [3]. There are three switching state presents in ternary value logic system [19]. Although TVL most often refers to a system in which all whole numbers can be represented by the three switching states i.e 0, 1, and 2 and each ternary digit is denoted by trit [3][19]. In TVL system amount of information can be stored more than binary system. TVL system has some capabilities to enhance processor capacity, accurate processing of signal, less memory required in compare with binary system [2][20]. The arithmetic operations of TVL Systems are Compliment Operation, Addition, Subtraction, multiplication and division [21][22].

2.2 Review Of Double Base Ternary Number Systems (DBTNS)

An integer can be represented as a sum of mixed powers of two integers, two (2) and three (3) respectively. Mathematically representing technique is termed as Double Base Number System i.e. DBNS [16][23][24]. In the Double-Base number system, an integer, x, can be represented in (1).

$$x = \sum_{i,j} d_{i,j} 2^i 3^j \,, \qquad \text{where } d_{i,j}=\{0,1\} \qquad (1)$$

Form (1), a given binary number can be converted into DBNS as number of (i, j) pair. These are also referred to as DBNS indices [18][25][22]. In the Double-Base number system, when x, is a ternary number then x can be expressed as in (2) which is well elaborated in [24].

$$x = \sum_{i,j} d_{i,j} 2^i 3^j \,, \qquad \text{where } d_{i,j}=\{0,1,2\}. \qquad (2)$$

These indices (i, j) are in ternary number system. So conversion of a ternary number into DBNS as number of (i, j) pair in TVL domain [19] can be termed as Double Base Ternary Number Systems (DBTNS). Table 1 is representing DBTNS table where trit length of indices (i, j) is 1 and in table 2, trit length of indices (i, j) is 2. Dynamic range for 1 trit is 91 and 2 trit is 5028751.

Table 1. DBTNS Table for i, j → 1 trit

i \ j	0	1	2
0	0001	0010	0100
1	0002	0020	0200
2	0011	0110	1100

Table 2:.DBTNS Table for i, j → 2 trit

i, \ j	00	01	02	10	11	12	20	21	22
00	0000000 0000001	0000000 0000010	0000000 0000100	0000000 0001000	0000000 0010000	0000000 0100000	0000000 1000000	0000001 0000000	0000010 0000000
01	0000000 0000002	0000000 0000020	0000000 0000200	0000000 0002000	0000000 0020000	0000000 0200000	0000000 2000000	0000002 0000000	0000020 0000000
02	0000000 0000011	0000000 0000110	0000000 0001100	0000000 0011000	0000000 0110000	0000000 1100000	0000001 1000000	0000011 0000000	0000110 0000000
10	0000000 0000022	0000000 0000220	0000000 0002200	0000000 0022000	0000000 0220000	0000000 2200000	0000002 2000000	0000022 0000000	0000220 0000000
11	0000000 0000121	0000000 0001210	0000000 0012100	0000000 0121000	0000000 1210000	0000001 2100000	0000012 1000000	0000121 0000000	0001210 0000000
12	0000000 0001012	0000000 0010120	0000000 0101200	0000000 1012000	0000001 0120000	0000010 1200000	0000101 2000000	0001012 0000000	0010120 0000000
20	0000000 0002101	0000000 0021010	0000000 0210100	0000000 2101000	0000002 1010000	0000021 0100000	0000210 1000000	0002101 0000000	0021010 0000000
21	0000000 0011202	0000000 0112020	0000000 1120200	0000001 1202000	0000011 2020000	0000112 0200000	0001120 2000000	0011202 0000000	0112020 0000000
22	0000000 0100111	0000000 1001110	0000001 0011100	0000010 0111000	0000100 1110000	0001001 1100000	0010011 1000000	0100111 0000000	1001110 0000000

2.3 Review Of Ternary Residue Number Systems (RNS)

One of the well-known non-weighted number systems is residue number system (RNS) [11]. A number, ***X***, can be represented in RNS [26] as ***X*** = (***x_1, x_2, x_3,…,x_N***) where ***x_i = X*** *modulo* ***m_i***; where ***x_i*** is the i-th residue digit, the i-th modulus is denoted by ***m_i*** and all ***m_i*** should be mutually prime numbers. The maximum number of different values can be represented by dynamic range, M, as show in (3).

$$M = \prod_{i=0}^{N} m_i \ \ and \ \ X < M \tag{3}$$

For signed RNS, any integer in the range of (***-M/2, M/2***], has a unique RNS N tuple representation can be shown by (4).

$$\begin{aligned} x_i &= (X \text{ modulo } m_i) \text{ for } X>0, \\ &= ((M-|X|) \bmod m_i) \text{ for } X<0 \end{aligned} \tag{4}$$

One of the major issue related RNS is to select moduli set. To get the maximum dynamic range, moduli set should be relative prime [12][13]. Irregular choice of unbalanced moduli sets prompts inefficient architectures wherein the biggest modulus is too much prevailing regarding both cost and execution. An example of a moduli set with good balance is {r^n - 2, r^n - 1, r^n} where r denotes radix or base. So, a good balanced modulus set for TVL is {3^n - 2, 3^n - 1, 3^n} [14].

3 Architecture of Proposed MAC Unit Using DBTNS Multiplier and TRNS Adder

DSP algorithms are based on sum of product. So, MAC unit is the best solution for implementing signal processing algorithm because multiplication and accumulation operation can be performed in single cycle by MAC unit [4][5][6]. Main building blocks of a conventional MAC unit are multiplier and accumulator which is used for storing the sum of the previous consecutive products [5][27]. So, multiplier, adder and accumulator are required for implementing MAC unit. Single cycle multiplication and accumulation can be done using this MAC unit [5]. In the architecture of the proposed MAC unit, there are two input h(n) and x(n) which is in TVL. Initially, they are converted into DBTNS and x(n) and h(n) are multiplied using DBTNS multiplier. Then the TVL based product is converted in RNS to perform the carry free addition in TVL domain. The proposed architecture of DBTNS - TRNS mixed base Multiply-Accumulate (MAC) Unit is depicted in the figure 1. So, the following modules are required for implementation of proposed DBTNS - TRNS mixed base MAC unit.

A. Integer to TVL Conversion
B. DBTNS Conversion
C. DBTNS Multiplier
D. TVL – RNS Conversion
E. TRNS Adder
F. Ternary Registrar (TReg)

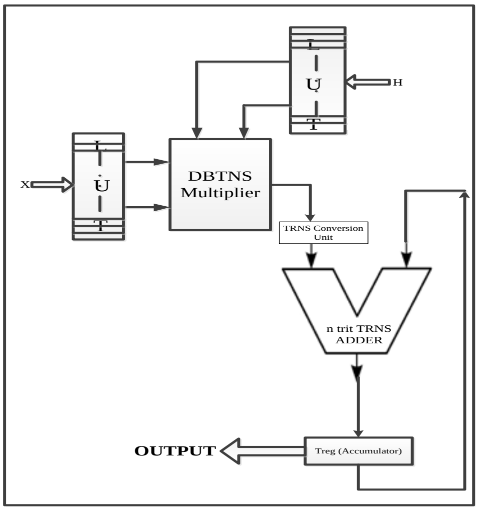


Figure 1. DBTNS-TRNS mixed base MAC Unit

3.1 Integer to TVL Conversion Unit

The conversion of Integer to Ternary Value Logic is carried out by this unit. The approach is totally Look Up Table i.e. LUT [22][28] based. The approach is depicted in figure 2.

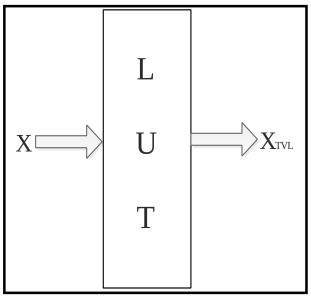


Figure 2. LUT Based TVL Conversion Unit

3.2 DBTNS Conversion Unit

The conversion of Ternary Number to Double Base Ternary Number System is carried out by this unit. The approach is totally Look Up Table (LUT) based [23]. In DBTNS, there are two bases, one is 2 and another is 3 and the number is represented in terms of power 2 and 3 i.e $X=2^{i}.3^{j}$ where these indices (i, j) are in ternary number system [19][28]. Here, the values of i and j are stored in different location of LUT as shown in figure 3. These i and j are used in the consecutive steps.

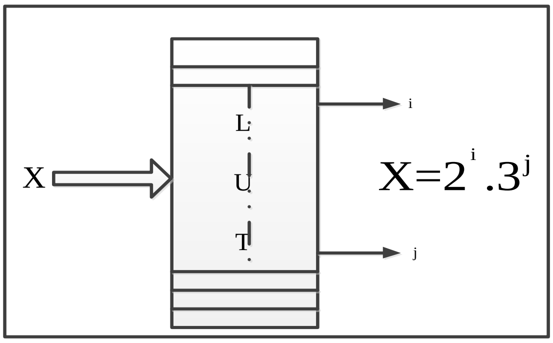


Figure 3. LUT Based DBTNS Conversion Unit

3.3 DBTNS Multiplier Unit

Suppose, X_1 and X_2 are two ternary numbers. In DBTNS, $X_1 = 2^{i_1}.3^{j_1}$ and $X_2 = 2^{i_2}.3^{j_2}$. Now, Z = X_1. X_2 then $Z = 2^{(i_1+i_2)}.3^{(j_1+j_2)}$. The architecture of DBTNS Multiplier [17][28] is depicted in the figure 4. The operation of DBTNS is vividly discussed in [24]. In the architecture of DBTNS multiplier [24], 'n' is the trit length of indices, 'N' is the trit length of ternary equivalent of power of 2 i.e. $2^{(i_1+i_2)}$ and 'M' is the trit length of product. For implementing this architecture, Tenrany Adder, Barrel Shifter and LUT are required [24][28][29]. Ternary Ripple Carry Adder (TRCA) is used to add the indices. The ternary equivalent of power of 2 i.e. $2^{(i_1+i_2)}$ is kept in the LUT. This stored data is passed through a barrel shifter as it has ability to perform multi-trit shifting in a single cycle. The amount of shift is defined by the power of 3 i.e. (j_1 + j_2). The multiplied result can be collected from the barrel shifter. Performance analysis of DBTNS multiplier is depicted in the table 3 [24].

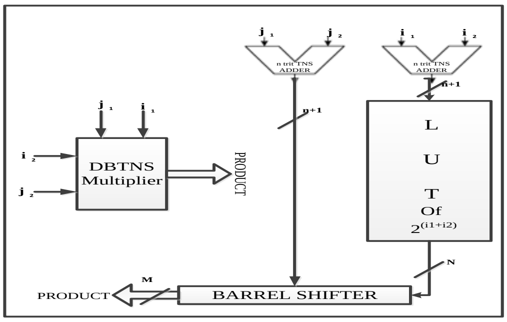


Figure 4. DBTNS Multiplier Unit

Table 3. Data Table of DBTNS Multiplier

INPUT INDEX TRIT LENGTH (i, j) (n)	TNS ADDER		LUT DATA RANGE	BARREL SHIFTER INPUT DATA TRIT LENGTH (N)	MAXIMUM SHIFT	PRODUCT (M)
	OUTPUT TRIT LENGTH (n+1)	OUTPUT DATA RANGE				
1	2	00 to 11	2^0 to 2^4	3	4	7
2	3	000 to 121	2^0 to 2^{16}	11	16	27
3	4	0000 to 1221	2^0 to 2^{52}	33	52	85

3.4 TRNS Conversion Unit

A ternary number X can be represented by residue number system (RNS) in TVL domain [14]. A number, *X*, can be represented in RNS [26] as $X = (x_1, x_2, x_3,...,x_N)$ where $x_i = X \text{ modulo } m_i$; where x_i is the i-th residue digit, the i-th modulus is denoted by m_i and all m_i should be mutually prime numbers. The maximum number of different values can be represented by dynamic range, M, which can be expressed by equation (3). The moduli set {r^n - 2, r^n - 1, r^n} are selected for the proposed architecture where r denotes radix or base and for TVL system, the value of r is 3 and n = 1, 2, 3 4 trit. X is a 2n trit TVL number. When TRNS conversion for modulus 3^n is performed on X, the generated output is X_{m0} and the computed output is as X_{m0} = X(n-1 downto 0). The TRNS conversion for modulus 3^n is shown figure – 5. The TRNS conversion for modulus (3^n – 1) is shown figure 6. Initially, n trit least significant digit is added with n trit most significant digit. As considered earlier, X is a 2n trit TVL number. The operation is represented by equation (5).

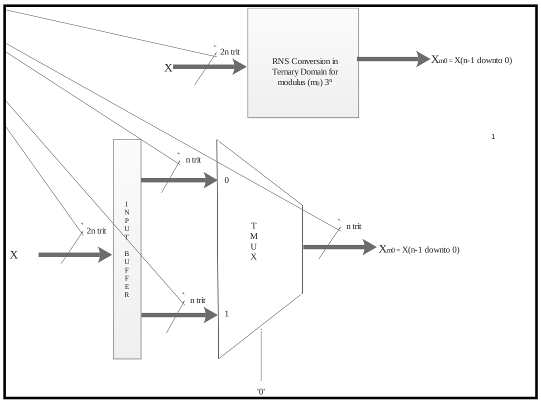


Figure 5. TRNS conversion for modulus 3^n

$$
\begin{aligned}
X_{m1} &= X(2n\text{-}1 \text{ downto } n) + X(n\text{-}1 \text{ downto } 0) && \text{when carry out} = 0;\\
&= X(2n\text{-}1 \text{ downto } n) + X(n\text{-}1 \text{ downto } 0) + \text{‘1’} && \text{when carry out} = 1;\\
&= X(2n\text{-}1 \text{ downto } n) + X(n\text{-}1 \text{ downto } 0) + \text{‘2’} && \text{when carry out} = 2; \qquad (5)
\end{aligned}
$$

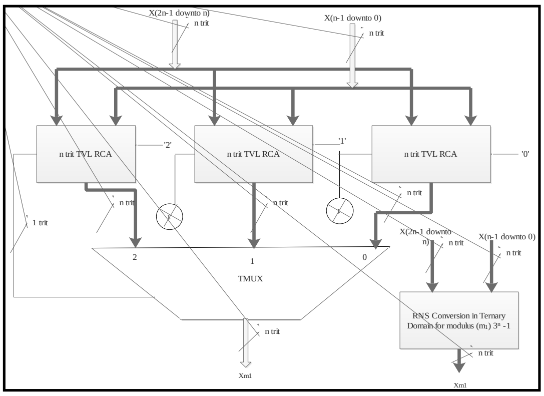


Figure 6. TRNS conversion for modulus 3^n-1

The TRNS conversion for modulus (3^n – 2) is shown figure 7. The operation of this module is described by equation (6).

$$
\begin{aligned}
X_{m2} &= \{2X(2n\text{-}1 \text{ downto } n)\} + X(n\text{-}1 \text{ downto } 0) && \text{when carry out} = 0;\\
&= \{2X(2n\text{-}1 \text{ downto } n)\} + X(n\text{-}1 \text{ downto } 0) + \text{‘2’} && \text{when carry out} = 1;\\
&= \{2X(2n\text{-}1 \text{ downto } n)\} + X(n\text{-}1 \text{ downto } 0) + \text{‘4’} && \text{when carry out} = 2; \qquad (6)
\end{aligned}
$$

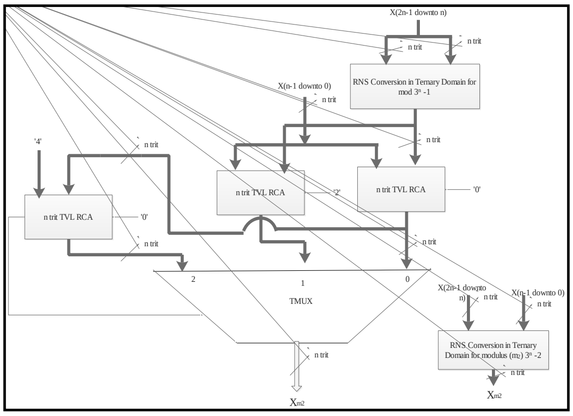


Figure 7. TRNS conversion for modulus 3^n-2

3.5 Multi-trit TRNS Adder

The multi-trit TRNS adder [14][26] is implemented based on carry free RNS adder in ternary value logic (TVL). TRNS adder is implemented using ternary adder, ternary subtractor, ternary logic comparator and ternary multiplexer. TRNS adder is shown in figure 8. To perform RNS addition in ternary value logic can be performed as in equation (7).

$$SUM_{RNS} = \begin{cases} A_i + B_i & \text{if } A_i + B_i < m_i \\ A_i + B_i - m_i & \text{otherwise} \end{cases} \qquad (7)$$

where A_i, B_i are n trit input data and m_i is the is the i-th modulus and all m_i are mutually prime numbers. Whether the added data (A_i + B_i) is lesser or greater that decision is made by logic comparator. Depending on the decision of logic comparator, operation of demux and mux is decided which one have to pass and which have to restrict. Trit length analysis of TRNS adder is described in the table 4 for different trit length of moduli.

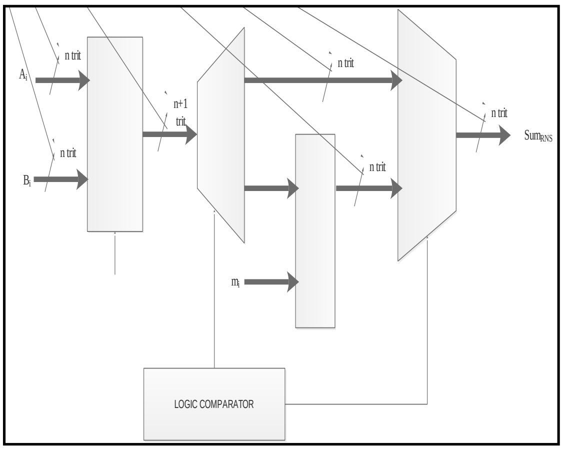


Figure 8. TRNS Adder

Table 4. Data Table of TRNS Adder

INPUT TRIT LENGTH (n)	TVL ADDER		MODULI SET $\{r^n - 2, r^n - 1, r^n\}$ r: RADIX = 3 (for TVL)	DYNAMIC RANGE (M)	OUTPUT TRIT LENGTH (same as input trit length i.e. n)
	OUTPUT TRIT LENGTH (n+1)	OUTPUT DATA RANGE			
1	2	00 to 11	{1, 2, 3}	6	1
2	3	000 to 121	{7, 8, 9}	504	2
3	4	0000 to 1221	{25, 26, 27}	17,550	3
4	5	00000 to 122221	{79, 80, 81}	5,11,920	4

4 Principle Of Operation Of DBTNS - TRNS Mixed Base MAC Unit For FIR Filter

Multiply-accumulate operation can be performed in single cycle by MAC unit [4][5]. The architecture of the proposed MAC unit is shown in figure 9. There are two inputs, h(n) & x(n) in a MAC unit. Initially they are converted into TVL system of n trit and these TVL converted values are passed through DBTNS converter to generate indices. Then these inputs are passed through DBTNS multiplier to perform multiplication. In the proposed architecture, there are 5 (five) LUTs, among these 5 LUTs, LUT-1 & LUT-

3 are used for integer to ternary number, LUT-2 & LUT-4 are used for converting TNS to double base ternary number and LUT-5 is used to convert ternary to integer. Initially, *x(n)* and *h(n)* are converted into DBTNS i.e. $x(n) = 2^{i_1}.3^{j_1}$ and $h(n)= 2^{i_2}.3^{j_2}$. The indices of 2 & 3 are passed through DBTNS Multiplier unit and multiplied data is added with zero which is initially stored in an accumulator. Trit length analysis of DBTNS multiplier is cited in the table – 3. Then, this multiplied data is converted into ternary RNS i.e. TRNS of n trit. The moduli set {3^n - 2, 3^n - 1, 3^n} are selected for the proposed architecture and n = 1, 2, 3 4 trit. In the proposed architecture, there are 3 (three) LUTs, among these 3 LUTs, LUT-1 & LUT-2 are used for integer to ternary number and LUT-3 is used to convert TRNS to TVL. The multiplied data in TVL is denoted by X. After TRNS conversion X is represented by {X_{m2}, X_{m1}, X_{m0}}. Here LUT is used for storing the inputs of FIR filter [30][31][32]. Here we are considering the number of tapping is four. FIR algorithm can be implemented using MAC unit. FIR filter can be represented by the equation (8).

$$y(n)=\sum_{k=0}^{N-1} x(n-k).h(k) \qquad (8)$$

The operations of FIR Algorithm are taken place as follows:

a) To check FIR filter algorithm, filter coefficients (h(n)) are stored in a LUT in TVL system. Another input, x(n) is given from another source. x(n) is also in TVL system.
b) When Program Counter (PC) [6] gets clock, its starts to point an address of LUT, then a data (i.e h(n)) sends for performing arithmetic operation with x(n).
c) The multiplied results are converted in TRNS depending on the moduli set {3^n - 2, 3^n - 1, 3^n} where n = 1, 2, 3, 4. Trit length analysis of TRNS adder is cited in the table 4 for different trit length of moduli.
d) RNS addition is performed on these TRNS data.
e) Initially, Ternary Registrar (TReg) [33] data is zero then TReg starts updating from the very next clock.
f) After completion of operation, stored data of TReg is converted to TVL data.

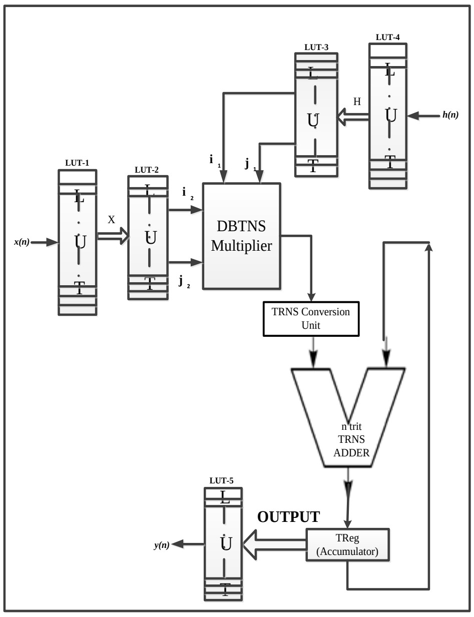


Figure 9. Block diagram of a proposed DBTNS - TRNS mixed base MAC unit

5 Performance Analysis of Proposed DBTNS - TRNS Mixed Base MAC Unit

DBTNS – TRNS based MAC unit is implemented using TVL conversion Unit, DBTNS conversion unit, DBTNS Multiplier Unit, TRNS Conversion Unit, TRNS Adder Unit and Ternary Registrar Unit. The TVL conversion is performed by LUT [3][8] based approach. So over all time complexity [6][20] depends on the LUT size. To implement FIR filter using TRNS MAC unit, total delay can be represented as

(n - trit LUT access time for integer to TVL conversion + n - trit LUT access time for TVL to DBTNS conversion + time taken by DBTNS multiplier + time taken by TRNS conversion + time taken by TRNS Adder + n - trit LUT access time for TNS to integer conversion).

DBTNS multiplier is implemented using ternary Ripple Carry Adder (TRCA) and ternary barrel shifter. Due to single cycle multitrit shifting ability, choosing of barrel shifter can perform. So the time taken by DBTNS multiplier can be calculated as

(time taken by n – trit ternary RCA + time taken by ternary Barrel Shifter)

TRNS Conversion Unit is implemented using ternary Ripple Carry Adder (TRCA) and ternary multiplexer. So the time taken by TRNS conversion unit [14] can be calculated as

(time taken by n – trit ternary RCA + time taken by ternary Multiplexer).

TRNS Adder Unit is implemented using ternary Ripple Carry Adder (TRCA), ternary subtractor and ternary multiplexer and de-multiplexer. So time taken by TRNS Adder Unit can be calculated as

(time taken by n – trit TRCA + time taken by ternary De-multiplexer + time taken by n – trit ternary Subtractor + time taken by ternary Multiplexer).

If the number of trit of input data of FIR filter [4][27][29][30] are changed then execution time is also varied. The synthesis report of 8 tap FIR filter with change of trit is shown in the table 5. The relation between number of LUTs and number of trit, maximum frequency and trit and execution time and trit is shown in the figure 10.

Table 5. Synthesis report of 8 tap FIR filter using DBTNS - TRNS mixed based MAC unit with change of Trit

Sl. No.	Moduli	Input Index Trit Length (i, j)	Synthesis Report				
			Minimum period	Maximum Frequency	Minimum input arrival time before clock	Maximum output required time after clock	Number of Slice LUTs
1.1	7, 8, 9	1	2.439ns	410.071MHz	7.399ns	0.609ns	169 out of 46560
1.2		2	2.439ns	410.071MHz	9.525ns	0.609ns	203 out of 46560
1.3		3	2.439ns	410.071MHz	9.805ns	0.609ns	209 out of 46560
2.1	25, 26, 27	1	2.775ns	360.386MHz	9.390ns	0.595ns	224 out of 46560
2.2		2	2.775ns	360.386MHz	10.979ns	0.595ns	269 out of 46560
2.3		3	2.775ns	360.386MHz	11.809ns	0.595ns	273 out of 46560
3.1	79, 80, 81	1	3.232ns	309.377MHz	11.875ns	0.604ns	285 out of 46560
3.2		2	3.232ns	309.377MHz	12.770ns	0.604ns	366 out of 46560
3.3		3	3.232ns	309.377MHz	13.504ns	0.604ns	379 out of 46560

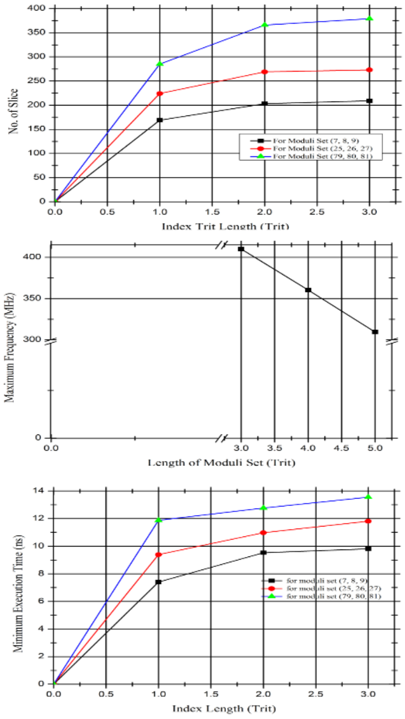


Figure 10. Complexity analysis of 8-tap FIR Filter using DBTNS – TRNS mixed base MAC unit with the change of trit

6 Conclusion

The multivalued logic approach i.e TVL which offers several benefits over existing binary digital system [3][24][34][35]. In this paper, a new architecture for MAC units in ternary value logic (TVL) domain has been proposed for implementing DSP algorithm like FIR algorithm [20][27] using DBTNS multiplier and TRNS adder. TRNS adder can perform carry free addition in ternary domain efficiently and partial product free multiplication operations can be performed by DBTNS multiplier efficiently. Since TRNS adder is efficient compared to conventional TVL adder and DBTNS multipliers are efficient compared [6][28] to conventional ternary multiplier. The novelty of the proposed MAC unit is justified by analyzing the experimental results. The architecture was validated on Xilinx FPGA [18][36][37] and the detailed analysis and studies of different modules of the proposed units have been simulated using Xilinx ISE version 12.3. A detailed study shows improvements on other DSP algorithms [4] like speech processing, high quality sound systems, adaptive echo cancellation, solar signal processing, military applications etc where in addition to high speed, high precisions are also required [34][38] [39].

REFERENCES

[1] Chung-Yu-Wu., "Design& application of pipelined dynamic CMOS ternary logic & simple ternary differential logic" IEEE journal on solid state circuits, 28: 895-906, 1993.

[2] S.L. Hurst, "Multiple-valued logic--its status and its future," IEEE Transactions on Computers, vol. C-33, no. 12, pp. 1160-1179, December 1984.

[3] Reto Zimmermann "Lecture notes on Computer Arithmetic: Principles,Architectures and VLSI Design," Integrated System Laboratory, Swiss Federal Institute of Technology (ETH) Zurich, Mar,16, 1999. URL http://www.iis.ee.ethz.ch/zimmi/publications/comp_arith notes.ps.gz

[4] Sanjit K.Mitra "Digital Signal Processing",A Wiley-Inter science Publication,1999.

[5] Kai Hwang (Purdue University) and Faye A. Briggs (Rice University), "COMPUTER ARCHITECURE AND PARALLEL PROCESSING", International Edition 1985.

[6] J. P. Hayes, "Computer Organization", (3rd edition), McGraw-Hill, 1998.

[7] R.Mariani, F.Pessolano & R.Saletti ' A new CMOS ternary logic design for low power low voltage circuit' Tutorial University of Pisa, Italy.

[8] Radanovic M. Syrzycki., "Current-mode CMOS adders using multiple-valued logic", Canadian Conference on Electrical and Computer Engineering, 190-193, 1996.

[9] Gonzalez F, Mazumder P., "Multiple-valued signed digit adder using negative differential resistance devices." IEEE Trans. on Computers. 47: 947 - 959, 1998.

[10] Wei Wang, M.N.S. Swamy, and M.O. Ahmad, "Modulii Selection in RNS for Efficient VLSI Implementation", IEEE Press, New York, pages. IV-512 ~ 515, May, 2003.

[11] Chao-Lin Chiang and Lennart Johnsson, "Residue Arithmatic and VLSI" Presented at 1983 IEEE Internal Conference on Computer Design: VLSI in Computers (ICCCD'83), New York, Oct. 31 - Nov 3, 1983.

[12] Eep Setiaarif, Pepe Siy, "A New Modulii Set Selection Technique To Improve Sign Detection And Number Comparison In Residue Number System (RNS)", NAFIPS 2005 - 2005 Annual Meeting of the North American Fuzzy Information Processing Society, IEEE Press, New York, pages 766 ~ 768, 2005.

[13] Abdallah, M. and A. Skavantzos, "On multi moduli residue number systems with moduli of forms {ra, rb-1,rc+1}", IEEE Trans. Circuits Syst. I: Regular Paper, Vol.52, pages 1253-1266, 2005.

[14] M. Hosseinzadeh and K. Navi, "A New Moduli Set for Residue Number System in Ternary Valued Logic", Journal of Applied Sciences, Vol. 7(23), pages 3729-3735, 2007.

[15] Mandyam S. and Stouraitis T. "Efficient Analog to Residue Conversion Schemes". IEEE International Symposium on Circuits and Systems. New Orleans, USA, pages 2885-2888, May 1990.

[16] V. S. Dimitrov, G. A. Jullien, W. C. Miller, Theory and Applications of the Double-Base Number System. IEEE Trans. Computers, Vol. 48, 10, pp.1098-1106, 1999.

[17] Satrughna Singha, Aniruddha Ghosh and Amitabha Sinha, "A New Architecture for FPGA based Implementation of Conversion of Binary to Double Base Number System (DBNS) Using Parallel Search Technique", ACM SIGARCH Computer Architecture News, Volume 39, Issue 5, pp. 12-18, ACM New York, USA, December 2011. DOI:10.1145/2093339.2093343.

[18] R. Tessier and W. Burleson, Reconfigurable computing for digital signal processing: A survey, Journal of VLSI Signal Processing, Vol 28, no.7-27, pp 7-27, 2001.

[19] Yoeli M, Rosenfeld G., "Logical Design of ternary switching circuits." IEEE Trans Computer., 14: 19-29, 1965.

[20] Parhami, B., "Computer Arithmetic: Algorithms and Hardware Designs", 1st Edn., Oxford University Press, Oxford, UK., 2001, ISBN: 0-19-512583-5.

[21] K.C.Smith 'Multiple Valued Logic: A tutorial & application'IEEE Tran. Computer Vol.21 p.p.17-21April 1988.

[22] Satrughna Singha and Amitabha Sinha, "Survey of Various Number Systems and Their Applications", International Journal of Computer Science and Communication, Volume-1, Number-1, pp. 73-76, Serials Publications, Kurukshetra University, Haryana, India, January 2010.

[23] V. S. Dimitrov, S. Sadeghi-Emamchaie, G. A. Jullien, W. C. Miller, A Near Canonic Double-Based Number System (DBNS) with Applications in Digital Signal Processing. Proceedings SPIE Conference on Advanced Signal Processing, August, 1996.

[24] Aniruddha Ghosh and Amitabha Sinha, "FPGA Implementation of MAC Unit for Double Base Ternary Number System (DBTNS) and its Performance Analysis", International Journal of Computer Applications, Volume 181, Issue 14, pp. 9-22, September 2018, Foundation of Computer Science (FCS), NY, USA, DOI:10.5120/ijca2018917785.

[25] R. Muscedere, V. S. Dimitrov, G. A. Jullien, W. C. Miller, M. Ahmadi, On Efficient Techniques for Difficult Operations in One and Two-digit DBNS Index Calculus. Proceedings 34th Asilomar Conference on Signals, Systems and Computers, November, 2000.

[26] Fred J. Taylor, "Residue Arithmetic: A Tutorial with Examples", IEEE Trans. on Computer, pp. 50~62, May 1984.

[27] Conway, R. and J. Nelson, “Improved RNS FIR filter architectures. IEEE Trans”, Circuits Syst. II: Express Briefs, Vol. 51, pages 26-28, 2004.

[28] Aniruddha Ghosh, Satrughna Singha and Amitabha Sinha, “A New Architecture for FPGA Implementation of A MAC Unit for Digital Signal Processors using Mixed Number System”, ACM SIGARCH Computer Architecture News, Volume 40, Issue 2, pp. 33-38, ACM New York, USA, May 2012. DOI:10.1145/2234336.2234342.

[29] A. Sinha, P. Sinha, K. Newton, K. Mukherjee, Multi based number systems for performance enhancement of Digital Signal Processors. Filed for U.S. patent. (U.S. Pat. Appl. No. 11/488,138) ,published in U.S. Patent documents serial no.488138 ,U.S. Class at publication 708/620, int'l class : G06F 7/52 20060101 G06F007/52, 2006.

[30] J. Eskritt, R. Muscedere, G. A. Jullien, V. S. Dimitrov, W. C. Miller, A 2-Digit DBNS Filter Architecture. Proceedings SiPS Workshop (Lafayette, L A), October, 2000.

[31] G. A Jullien, V. S. Dimitrov, B. Li, W. C Miller, A. Lee, M. Ahmadi, A Hybrid DBNS Processor for DSP Computation. Proceedings International Symposium on Circuits and Systems, 1999.

[32] Nevio Benvenuto, Lewis E. Franks, and F. S. Hill, Jr. “Realization of Finite Impulse Response Filters Using Coefficients +1, 0 and -1” IEEE Transactions on Communications, vol. COMM-33, no. 10, October 1985.

[33] Dhande P, Ingole VT., "Design of clocked ternary S-R and D flip-flop based on simple ternary gates.", International journal on software engineering and knowledge engineering, 15: 411417, 2005.

[34] Gonzalez F, Mazumder P., "Multiple-valued signed digit adder using negative differential resistance devices." IEEE Trans. on Computers. 47: 947 - 959, 1998.

[35] Radanovic M. Syrzycki., "Current-mode CMOS adders using multiple-valued logic", Canadian Conference on Electrical and Computer Engineering, 190-193, 1996.

[36] Alireza Kaviani and Stephen Brown,"HYBRID FPGA ARCHITECTURE", FPGA.96, Monterey, CA, Feb.1996, pp. 1-7.

[37] C.Rozon ‘On the use of VHDL as a Multivalued Logic Simulator’ Proc. ISMVL 1996,p.p.110-115.

[38] B. G. Lee, A new Algorithm to compute the discrete cosine transforms. IEEE Trans on Acoustics, speech and signal Processing, vol. ASSP-32, pp.1243- 1245, December, 1984.

[39] Ping Wah Wong, “Fully Sigma-Delta Modulation Encoded FIR Filters” IEEE Transactions on Signal Processing, vol. 40, no. 6, pp. 1605-1610, June 1992.