ISSN:2229-6093

Aman Chadha et al, Int. J. Comp. Tech. Appl., Vol 2 (5), 1419-1425

Rotation, Scaling and Translation Analysis of Biometric Signature Templates

Aman Chadha, Divya Jyoti, M. Mani Roja
Thadomal Shahani Engineering College, Mumbai, India
aman.x64@gmail.com

Abstract

Biometric authentication systems that make use of signature verification methods often render optimum performance only under limited and restricted conditions. Such methods utilize several training samples so as to achieve high accuracy. Moreover, several constraints are imposed on the end-user so that the system may work optimally, and as expected. For example, the user is made to sign within a small box, in order to limit their signature to a predefined set of dimensions, thus eliminating scaling. Moreover, the angular rotation with respect to the referenced signature that will be inadvertently introduced as human error, hampers performance of biometric signature verification systems. To eliminate this, traditionally, a user is asked to sign exactly on top of a reference line. In this paper, we propose a robust system that optimizes the signature obtained from the user for a large range of variation in Rotation-Scaling-Translation (RST) and resolves these error parameters in the user signature according to the reference signature stored in the database.

Keywords: *rotation; scaling; translation; RST; image registration; signature verification.*

1. Introduction

The aim of a biometric verification system is to determine if a person is who he/she purports to be, based on one or more intrinsic, physical or behavioral attributes. This trait or biometric attribute can be the signature, voice, iris, face, fingerprint, hand geometry etc.

A simple biometric system has a sensor module, a feature extraction module, a matching module and a decision making module. The sensor module acquires the biometric data of an individual. In this case, the digital pen tablet functions as the sensor. In the feature extraction module, the acquired biometric data is processed to extract a feature set that represents the data. For example, the position and orientation of certain specific points in a signature image are extracted in the feature extraction module of a signature authentication system. In the matching module, the extracted feature set is compared against that of the template by generating a matching score. In this module, the number of matching points between the acquired and reference signatures are determined, and a matching score is obtained. Decision-making involves either verification or identification. In the decision-making module, the user's claimed identity is either accepted or rejected based on the matching score, i.e., verification. Alternately, the system may identify a user based on the matching scores, i.e., identification [1],[11].

Signature recognition is one of the oldest biometric authentication methods, with wide-spread legal acceptance. Handwritten signatures are commonly used to approbate the contents of a document or to authenticate a financial transaction [1]. A trivial method of signature verification is visual inspection. A manual comparison of the two signatures is done and the given signature is accepted if it is sufficiently similar to the reference signature, for example, on a credit-card. In most scenarios, where a signature is used as the means of authentication, no verification takes place at all due to the entire process being excessively time intensive and demanding. An automated signature verification process will help improve the current situation and thus, eliminate fraud. Well-known biometric methods include iris, retina, face and fingerprint based identification and verification. Even though human features such as iris, retina and fingerprints do not change over time and have low intra-class variation, i.e., the variations in the respective biometric attribute are low, special and relatively expensive hardware is needed for data acquisition in such systems. An important advantage of signatures as the human trait for biometric authentication over other attributes is their long standing tradition in many commonly encountered verification tasks. In other words, signature verification is already accepted by the general public. In addition, it is also relatively less expensive than the other biometric methods [1],[2].

The difficulties associated with signature verification systems due to the extensive intra-class variations, make signature verification a difficult pattern recognition problem. Examples of the various alterations observed in the signature of an individual have been illustrated in Fig. 1.

Figure 1: Intra-class variations, i.e. variations in the signature of an individual

Depending on the data acquisition method, automatic signature verification can be divided into two main types: off-line and on-line signature verification. The most accurate systems almost always take advantage of dynamic features like acceleration, velocity and the difference between up and down strokes [3]. This class of solutions is called on-line signature verification. However in the most common real-world scenarios, because such systems require the observation and recording off the signing process, this information is not readily available. This is the main reason, why static signature analysis is still in focus of many researchers. On-line signature verification uses special hardware, such as a digitizing tablet or a pressure sensitive pen, to record the pen movements during writing. In addition to shape, the dynamics of writing are also captured in on-line signatures, which is not present in the 2-D representation of the signature and hence it is difficult to forge. Off-line methods do not require special acquisition hardware, just a pen and a paper, and are therefore less invasive and more user friendly. In the past decade a bunch of solutions has been introduced, to overcome the limitations of off-line signature verification and to compensate for the loss of accuracy [2],[3]. In off-line signature verification, the signature is available on a document which is scanned to obtain its digital image. In all applications where handwritten signatures currently serve as means of authentication, automatic signature verification can be used such as cashing a check, signing a credit card transaction or authenticating a legal document. Basically, any system that uses a password can instead use an on-line signature for access. The advantages are such systems are obvious – a signature is more difficult to steal or guess than a password and is also easier to remember for the user.

However, the high level of intra-class variations in signatures, as shown in Fig. 1, hinder the performance of signature verification systems and thus minimize the accuracy of such systems. Hence, to reduce errors and the inefficiency problems associated with these systems, the intra-class variations in the signatures need to be minimized. This involves eliminating or reducing the rotation, scaling and translation factors between the reference and the test signature images. Fig. 2 shows the diagram of a typical signature verification system with rotation, scaling and translation (RST) cancellation. The reference image within the database and the user image act as inputs to the system. Feature extraction is done from the reference signature which describes certain characteristics of the signature and stored as a template. For verification, the same features are extracted from the test signature and compared to the template.

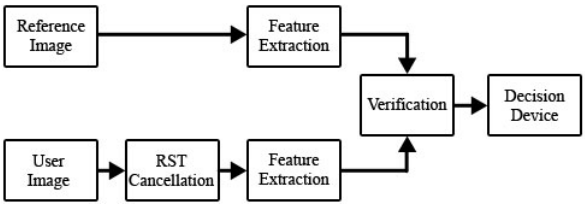


Figure 2: A typical signature verification system with RST cancellation

It should be noted that a distinct advantage of the proposed system, illustrated in Fig. 2, is that it does not require multiple signature reference samples for training in order to achieve high levels of accuracy. Previous work by researchers has witnessed the use of affine transformation for calculating the angular rotation between two images, the scaling and translation [4]-[7].

In this paper, we propose the use of the concept of correlation to identify the rotation and a simple cropping method to eliminate scaling and translation, thereby creating an optimum template after subjecting the user image to RST correction.

2. Idea of the proposed solution

The foremost concern is fetching the angle of rotation between the user and the reference images. In order to achieve this, the concept of correlation is deployed. The term “correlation” is a statistical measure, which refers to a process for establishing whether or not relationships exist between two variables [8]. The maximum value of cross-correlation between the original, i.e., the reference image and the user image is found by means of repetitive iterations involving the calculation of the cross-correlation between the two images in question.

The proposed algorithm essentially finds the cross-correlation between original image and the user image. If X(m, n) is reference image and Y(m, n) is the user image then the cross-correlation *r* between X and Y is given by the following equation:

$$r = \sum_m \sum_n (X_{mn} - X_0)(Y_{mn} - Y_0) \quad (1)$$

Minimum value of r indicates dissimilarity of images and for the same image (autocorrelation) it will have a peak value so as to indicate 'maximum correlation'. X_0 and Y_0 represent mean of Image X and Y respectively.

We use normalized cross-correlation to simplify analysis and comparisons of coefficient values corresponding to the respective angular values. Min-max normalization is the procedure used to obtain normalized cross-correlation [9]. Min-max normalization preserves the relationships among the original data values. The normalization operation transforms the data into a new range, generally [0, 1]. Given a data set x_i, such that i = 1, 2, . . , n, the normalized value x' is given by the following equation:

$$x' = \frac{x - \min(x_i)}{\max(x_i) - \min(x_i)} \quad (2)$$

The second aim is to deal with the translation associated with the images. This is achieved by a simple cropping technique. Initially, we calculate the number of rows and columns bordering the signature pixels within the image. The image devoid of these rows and columns is extracted. The result is an image consisting of only the signature pixels. Additional background surrounding the image is thus eliminated.

Third factor is the scaling between the two images. For calculation of the scaling factor, the cropped images obtained during translation are utilized. The size of the reference image divided by the size of the user image gives the scaling ratio.

The proposed solution is illustrated by Fig. 3.

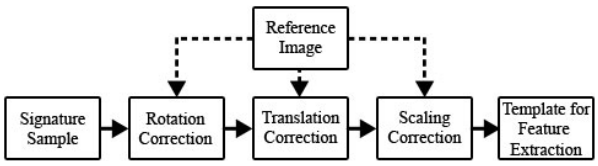


Figure 3: Schematic block diagram of the proposed system

3. Implementation steps

3.1. Image acquisition and pre-processing

Our image acquisition is inherently simple and does not employ any special illumination. The system implemented here uses a digital pen tablet, namely, WACOM Bamboo [10], as the data-capturing device. The pen has a touch sensitive switch in its tip such that only pen-down samples (i.e., when the pen touches the paper) are recorded. The database consists of a set of signature samples of 90 people. For each person, there are 9 test images and 1 training or reference image in the database. Upon signature acquisition, the next step is colour normalization and binarization. Colour normalization is the conversion of the image from the RGB form to the corresponding Grayscale image. Binarization is the conversion of this grayscale image to an image consisting of two luminance elements, namely, black and white. On completion of the image acquisition and pre-processing stage, the resultant image thus obtained, becomes ready for the corrective phases: rotation, scaling and translation cancellation.

Figure 4: Wacom Bamboo Digital Pen Tablet

3.2. Rotation correction

While collecting signature samples, it was observed that users gave consecutive samples having angular variations approximately from –60° to +60°. Hence, before feature extraction, the user image should be aligned with the reference image. For simplicity in computing rotation angle, we choose to align the reference image with the trial image fetched from the user, i.e., the user image.

The preprocessed reference image is cropped in order to extract only the signature pixels without any additional background and used for all further computations. In order to make the program time efficient and less resource intensive, two stages of rotation correction are applied. The first stage is designed to offer a relatively lower resolution of 5° so as to offer an approximate value of the angle of rotation. In contrast, the second stage is designed for a comparatively higher resolution. Within a range of +3° to –3° of the approximate value, a resolution of 1° is selected for a more precise value of the rotation angle.

After pre-processing, the user image is then rotated by 5° within the range of –60° to +60° in successive iterations. Cross-correlation values between the reference image and the user image are recorded on completion of each iteration of the rotation process. The maximum cross-correlation value refers to the correct angle of rotation within a 5° range, further, after the approximate angle value is obtained, +3° or –3° of this angle can be inspected for maximum correlation value which corresponds to angle of rotation accurate to up to 1°. The user image is rotated by the negative of

the angle thus obtained, and then subjected to feature extraction. Thus, rotation cancellation is achieved.

The steps involved in the rotation correction process can be summarized as follows:

1) *Obtain user image and the reference image.*
2) *Carry out pre-processing by converting both images to grayscale and performing normalization.*
3) *Trim the reference signature to remove any excess background; this will act as the template.*
4) *Starting with the angle as –60°, in increments of 5°, record normalized correlation values between pre-processed reference image and user image.*
5) *If angle is less than or equal to 60°, go to step 4.*
6) *Maximum correlation value corresponds to angle within a 5° range. Let this angular value be x°.*
7) *Starting with the angle as (x – 3)°, in increments of 1°, record normalized correlation values between the preprocessed reference image and the user image.*
8) *If angle is less than or equal to (x + 3)°, go to step 7.*
9) *Correct the user image by the obtained angle and proceed for further correction, if required.*

Fig. 5 shows a reference image and the corresponding image rotated by 20.9°.

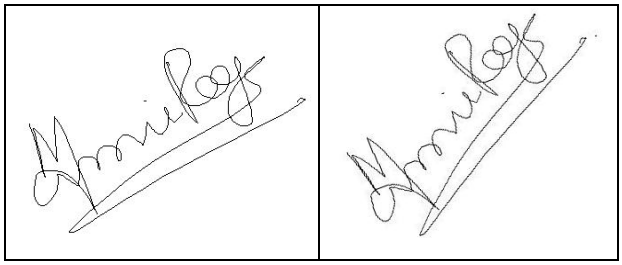


Figure 5: Reference image and the correponding rotated image

3.3. Scaling correction

An end user often modifies his/her signature according to the size of signing box. For smaller spaces, the signature may be compressed, for no space limitation, the sign may be enlarged. Thus, before extraction of feature points, it is essential that any scaling, if present in the test sample, be removed. Upon trimming both images, the ratio of height gives Y scaling and ratio of width gives X scaling. However, to resize the user image and make it the same size as the registered image, either of the scaling ratios can be used. For a rotation range of –60° to +60°, height was observed to vary significantly as compared to the length. Hence, Y scaling was chosen as the scaling ratio. To account for scaling, the above mentioned cropping technique is applied to both the user as well as reference image. Scaling ratio is calculated by the following equation:

$$\text{Scaling ratio} = \frac{\text{Size of the reference image}}{\text{Size of the test image}} \quad (3)$$

The user image is resized as per the obtained scaling ratio and then sent to the feature extraction segment. Fig. 6 shows a reference image and the corresponding image down scaled by a scaling ratio of 1.4045.

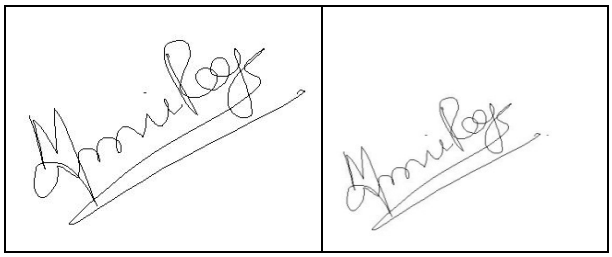


Figure 6: Reference image and the corresponding down-scaled image

3.4. Translation correction

On the apparatus used for taking signature input, the user is free to sign without using any fixed starting point. This may introduce translation in X and/or Y direction, having a maximum value equal to the width or height of the signature canvas respectively. The boundary conditions for translation error are computed assuming that the user starts to sign from the edge.

This problem is overcome by cropping the pre-processed reference image so as to extract only the signature pixels without any additional background. This cropping process truncates the extra background region by trimming the image canvas. Thus, translation is removed completely. For representational purposes, bottom left corner of test image is assumed to be origin.

The number of columns from left and number of rows from the bottom, which contain no black pixels corresponding to the actual signature, i.e., which consist solely of image-background, are counted. These values give X translation and Y translation respectively. Fig. 7 shows a reference image and the corresponding image translated by 35px along X-Axis and 9px along Y-Axis.

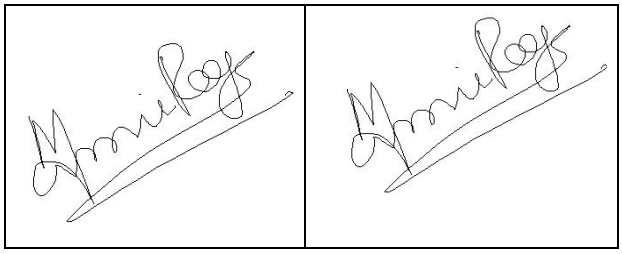


Figure 7: Reference image and the corresponding translated image

3.5. Combined Rotation–Scaling–Translation

It is easy to manipulate the samples to get pure rotation, translation and scaling, however, for actual signatures, all the above mentioned factors are altered simultaneously. Hence, rotation, translation and scaling corrections are applied in the same order.

Rotation correction precedes translation correction as the assumed origin at bottom left corner also gets rotated and translation effects cannot be eliminated unless the origin is returned to bottom left as accurately as possible. Therefore, rotation correction needs to be performed first as the scaling ratio calculated by the pure scaling method is not consistent with scaling ratio of the rotated image, as shown in Fig. 8.

Consequently, the effectiveness of scaling correction depends, to a large extent, on the percentage error obtained during rotation correction.

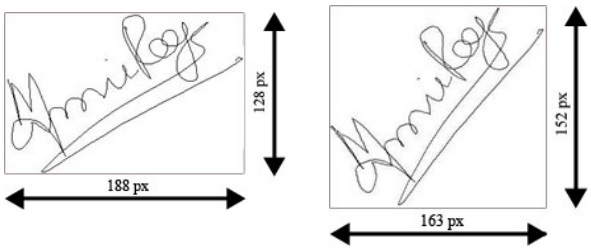


Figure 8: Change in the width and height of the image before and after rotation correction

After correcting the angle of rotation, the user image pre-processed copy is cropped to eliminate translation and the image so obtained is a case of pure scaling which has been discussed above.

Thus, rotation–scaling–translation cancellation is achieved.

4. Results

4.1. Rotation

Results obtained for pure rotation have been tabulated as follows:

Table 1: Results obtained for pure rotation

Signature Samples	Actual Angle	Detected Angle	% Error
Sample 1	-60	-60	0
Sample 2	-48	-48	0
Sample 3	-20	-20	0
Sample 4	-6	-6	0
Sample 5	0	0	0
Sample 6	4	4	0
Sample 7	13	13	0
Sample 8	27	27	0
Sample 9	37	37	0
Sample 10	59	60	1.69

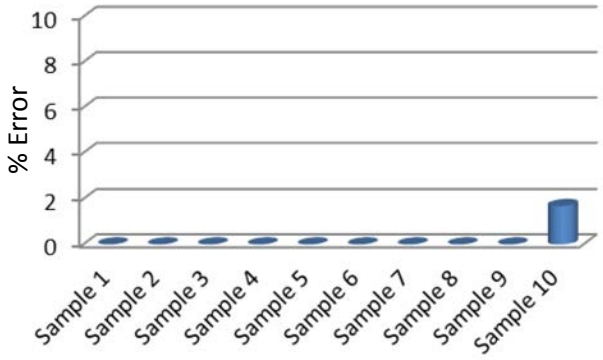


Figure 9: Plot of % error values in case of pure rotation for various samples

4.2. Scaling

Results obtained for pure scaling have been tabulated as follows:

Table 2: Results obtained for pure scaling

Signature Samples	Actual Scaling Ratio	Detected Scaling Ratio	% Error
Sample 1	7.69	10.55	37.2
Sample 2	5	5.70	14
Sample 3	4	4.22	5.5
Sample 4	2.17	2.27	4.6
Sample 5	1.28	1.34	4.7
Sample 6	1	1.05	5

Signature Samples	Actual Scaling Ratio	Detected Scaling Ratio	% Error
Sample 7	0.63	0.66	4.8
Sample 8	0.54	0.57	5.6
Sample 9	0.48	0.49	2.1
Sample 10	0.31	0.33	6.5

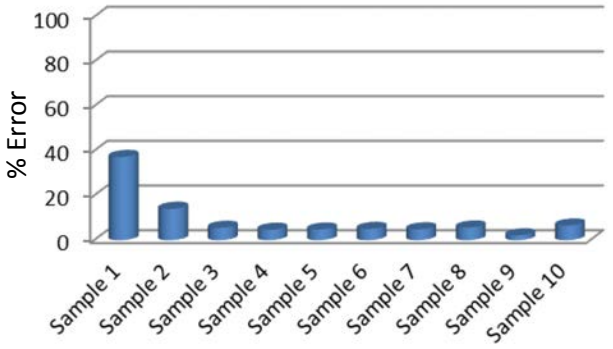


Figure 10: Plot of % error values in case of pure scaling for various samples

4.3. Translation

Results obtained for pure translation have been tabulated as follows:

Table 3: Results obtained for pure translation

Signature Samples	Actual Translation	Recovered Translation	% Error
Sample 1	0,5	0,5	0
Sample 2	5,5	5,5	0
Sample 3	10,0	10,0	0
Sample 4	15,10	15,10	0
Sample 5	0,25	0,25	0
Sample 6	25,25	25,25	0
Sample 7	25,50	25,50	0
Sample 8	50,50	50,50	0
Sample 9	50,100	50,100	0
Sample 10	150,150	150,150	0

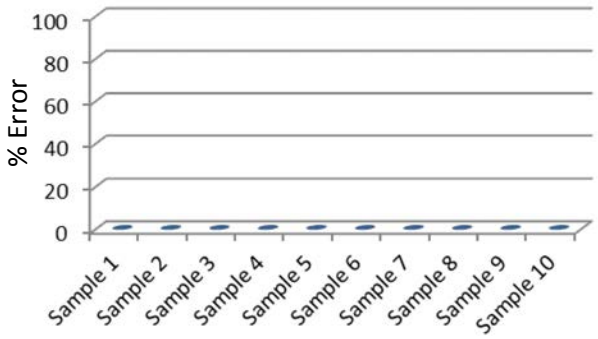


Figure 11: Plot of % error values in case of pure translation

4.4. Rotation-Scaling-Translation

Results obtained upon combining rotation, scaling and translation have been tabulated as follows:

Table 4: Results obtained on combining rotation, scaling and translation

Signature Samples	Actual Parameters		Detected Parameters	
	Rotation	*Scaling*	*Rotation*	*Scaling*
Sample 1	50	1.67	52	1.90
Sample 2	12	1.33	10	1.39
Sample 3	31	1.11	34	1.25
Sample 4	-40	0.91	-42	0.94
Sample 5	-30	0.8	-32	0.82

Figure 12: User image before RST correction (top); reference image (bottom-left) and the user image after RST Correction (bottom-right)

5. Conclusion

The system has been designed to correct variations in angle of rotation, in the range of –60° to +60°. However, it can be extended to cover the entire 360° planar rotation, to design a fool proof system capable of creating an optimum template after RST cancellation, even in a situation where the input pad may have been inverted. On similar lines, the resolution of rotation can be improved from 1° to 0.5° or 0.25° or even more, with the trade-off being increased program execution times.

Pure scaling and pure translation can be detected accurately as long as signature pixels do not go beyond the defining boundaries of the template. For the signatures used, a maximum translation of 200 pixels was detected along X and Y axes. Maximum scaling ratio was found to be 0.55. However, maximum variance of both translation and scaling may show slight variations from one signature to another.

For combined RST, it was experimentally observed that the correlation approach tends to be less reliable with significant increase or decrease in the scaling ratio. Signature images used for testing gave optimum result for scaling ratio, i.e., within 0.67 to 1.33, however, the scaling range giving angle and translation accurately may increase or decrease depending on the signature sample under test.

Thus, an optimum template was generated by the proposed system after subjecting the user image to RST correction with respect to the reference image.

6. References

[1] A. K. Jain, F. D. Griess and S. D. Connell, "On-line signature verification", *Elsevier*, Pattern Recognition 35, 2002, pp. 2963-2972.

[2] M. Mani Roja and S. Sawarkar, "A Hybrid Approach using Majority Voting for Signature Recognition", *International Conference on Electronics Computer Technology (ICECT) 2011*, pp. 1-3.

[3] B. Kovari, I. Albert and H. Charaf, "A General Representation for Modeling and Benchmarking Off-line Signature Verifiers", BME Publications, *12th WSEAS International Conference on Computers*, 2008, p. 1

[4] M. Holia and V. Thakar, "Image registration for recovering affine transformation using Nelder Mead Simplex method for optimization", *International Journal of Image Processing (IJIP)*, Volume 3, Issue 5, 2009. pp. 218-221.

[5] G. Wolberg and S. Zokai, "Robust Image Registration Using Log-Polar Transform", *Proceedings of the IEEE International Conference on Image Processing*, Sep. 2000 pp. 1-2.

[6] Z. Y. Cohen, "Image registration and object recognition using affine invariants and convex hulls", *IEEE Transactions on Image Processing*, July 1999, pp. 1-3.

[7] N. Chumchob and K. Chen, "A Robust Affine Image Registration Method", *International Journal Of Numerical Analysis And Modeling*, Volume 6, Number 2, pp. 311-334.

[8] R. J. Rummel, Understanding Correlation, Honolulu: Department of Political Science, University of Hawaii, 1976.

[9] J. Han, M. Kamber, *Data mining: concepts and techniques*, Morgan Kaufmann, 2006, pp. 70-72.

[10] WACOM Bamboo Digital Pen Tablet, www.wacom.co.in/bamboo, June 2011.

[11] Henk C. A. van Tilborg, *Encyclopedia of cryptography and security*, Springer, 2005, pp. 34-36.

CiiT International Journal of Digital Image Processing, ISSN 0974 – 9691 (Print) & ISSN 0974 – 9586 (Online)
Vol. 15, No. 10, October 2011

A robust, low-cost approach to Face Detection and Face Recognition

Divya Jyoti[1], Aman Chadha[2], Pallavi Vaidya[3], and M. Mani Roja[4]

***Abstract*—** In the domain of Biometrics, recognition systems based on iris, fingerprint or palm print scans etc. are often considered more dependable due to extremely low variance in the properties of these entities with respect to time. However, over the last decade data processing capability of computers has increased manifold, which has made real-time video content analysis possible. This shows that the need of the hour is a robust and highly automated Face Detection and Recognition algorithm with credible accuracy rate. The proposed Face Detection and Recognition system using Discrete Wavelet Transform (DWT) accepts face frames as input from a database containing images from low cost devices such as VGA cameras, webcams or even CCTV's, where image quality is inferior. Face region is then detected using properties of L*a*b* color space and only Frontal Face is extracted such that all additional background is eliminated. Further, this extracted image is converted to grayscale and its dimensions are resized to 128 x 128 pixels. DWT is then applied to entire image to obtain the coefficients. Recognition is carried out by comparison of the DWT coefficients belonging to the test image with those of the registered reference image. On comparison, Euclidean distance classifier is deployed to validate the test image from the database. Accuracy for various levels of DWT Decomposition is obtained and hence, compared.

***Keywords*—** discrete wavelet transform, face detection, face recognition, person identification.

I. Introduction

A face recognition system is essentially an application [1] intended to identify or verify a person either from a digital image or a video frame obtained from a video source. Although other reliable methods of biometric personal identification exist, for e.g., fingerprint analysis or iris scans, these methods inherently rely on the cooperation of the participants, whereas a personal identification system based on analysis of frontal or profile images of the face is often effective without the participant's cooperation or intervention. Automatic identification or verification may be achieved by comparing selected facial features from the image and a facial database. This technique is typically used in security systems. Given a large database of images and a photograph, the problem is to select from the database a small set of records such that one of the image records matched the photograph. The success of the method could be measured in terms of the ratio of the answer list to the number of records in the database. The recognition problem is made difficult by the great variability in head rotation and tilt, lighting intensity and angle, facial expression, aging, etc. A robust facial recognition system must be able to cope with the above factors and yet provide satisfactory accuracy levels. A general statement of the problem of machine recognition of faces can [2] be formulated as: given a still or video image of a scene, identify or verify one or more persons in the scene using a stored database of faces. The solution to the problem involves segmentation of faces, feature extraction from face regions, recognition, or verification. In identification problems, the input to the system is an unknown face, and the system reports back the determined identity from a database of known individuals, whereas in verification problems, the system needs to confirm or reject the claimed identity of the input face.

Some of the various applications of face recognition include driving licenses, immigration, national ID, passport, voter registration, security application, medical records, personal device logon, desktop logon, human-robot-interaction, human-computer-interaction, smart cards etc. Face recognition is such a challenging yet interesting problem that it has attracted researchers who have different backgrounds: pattern recognition, neural networks, computer vision, and computer graphics, hence the literature is vast and diverse. The usage of a mixture of techniques makes it difficult to classify these systems based on what types of techniques they use for feature representation or classification. To have clear categorization, the proposed paper follows the holistic approach [2]. Specifically, the following techniques are employed for facial feature extraction and recognition:

1) Holistic matching methods: These methods use the whole face region as a raw input to the recognition system. One of the most widely used representations of the face region is Eigenpictures, which is inherently based on principal component analysis.
2) Feature-based matching methods: Generally, in these methods, local features such as the eyes, nose and mouth are first extracted and their locations and local statistics are fed as inputs into a classifier.
3) Hybrid methods: It uses both local features and whole face region to recognize a face. This method could potentially offer the better of the two types of methods.

Manuscript received September 11, 2011.

[1] D. J. Rajdev is with the Thadomal Shahani Engineering College, Mumbai, 400002, INDIA (phone: +91-8879100684; e-mail: dj.rajdev@gmail.com).

[2] A. R. Chadha is with the Thadomal Shahani Engineering College, Mumbai, 400002, INDIA (phone: +91-9930556583; e-mail: aman.x64@gmail.com).

[3] P. P. Vaidya is with the Thadomal Shahani Engineering College, Mumbai, 400002, INDIA (e-mail: pallavi.p.vaidya@gmail.com).

[4] M. M. Roja is an Associate Professor in the Electronics and Telecommunication Engineering Department, Thadomal Shahani Engineering College, 400050, INDIA (e-mail: maniroja@yahoo.com).

Most electronic imaging applications often desire and require high resolution images. 'High resolution' basically means that pixel density within an image is high, and therefore a HR image can offer more details and subtle transitions that may be critical in various applications [19]. For instance, high resolution medical images could be very helpful for a doctor to make an accurate diagnosis. It may be easy to distinguish an object from similar ones using high resolution satellite images, and the performance of pattern recognition in computer vision can easily be improved if such images are provided. Over the past few decades, charge-coupled device (CCD) and CMOS image sensors have been widely used to capture digital images. Although these sensors are suitable for most imaging applications, the current resolution level and consumer price will not satisfy the future demand [19].

Past studies by researches and scientists that have investigated the challenging task of face detection and recognition have therefore, typically used high resolution images. Moreover, most standard face databases such as the MIT-CBCL Face Recognition Database [21], CMU Multi-PIE [22], The Yale Face Database [23] etc., that are basically used as a standard test data set by researchers to benchmark their results, also employ high quality images.

Results obtained by solutions proposed by researchers are therefore, relevant for theoretical understanding of face detection and identification in most cases. Practical conditions being rarely optimal, a number of factors play an important role in hampering system performance. Image degradation, i.e., loss of resolution caused mainly by large viewing distances as demonstrated in [4], and lack of specialized high resolution image capturing equipment such as commercial cameras are the underlying factors for poor performance of face detection and recognition systems in practical situations. There are two paradigms to alleviate this problem, but both have clear disadvantages. One option is to use super-resolution algorithms to enhance the image as proposed in [20], but as resolution decreases, super-resolution becomes more vulnerable to environmental variations, and it introduces distortions that affect recognition performance. A detailed analysis of super-resolution constraints has been presented in [3]. On the other hand, it is also possible to match in the low-resolution domain by downsampling the training set, but this is undesirable because features important for recognition depend on high frequency details that are erased by downsampling. These features are permanently lost upon performing downsampling and cannot be recovered with upsampling [24].

The proposed system has been designed keeping in view these critical factors and to address such bottlenecks.

II. Idea of the Proposed Solution

The database consists of a set of face samples of 50 people. There are 5 test images and 5 training or reference images. Frontal face images are detected and hence, extracted. DWT is applied to the entire image so as to obtain the global features which include approximate coefficients (low frequency coefficients) and detail coefficients (high frequency coefficients). The approximate coefficients thus obtained, are stored and the detail coefficients are discarded. Various levels of DWT are realized and their corresponding accuracy rates are determined.

A. Frontal Face Image Detection and Extraction

The face detection problem can be defined as, given an input an arbitrary image, which could be a digitized video signal or a scanned photograph, determine whether or not there are any human faces in the image and if there are, then return a code corresponding to their location. Face detection as a computer vision task has many applications. It has direct relevance to the face recognition problem, because the first and foremost important step of an automatic human face recognition system is usually identifying and locating the faces in an unknown image [5].

For our purpose, face detection is actually a face localization problem in which the image position of single face has to be determined [6]. The goal of our facial feature detection is to detect the presence of features, such as eyes, nose, nostrils, eyebrow, mouth, lips, ears, etc., with the assumption that there is only one face in an image [7]. The system should also be robust against human affective states of like happy, sad, disgusted etc. The difficulties associated with face detection systems due to the variations in image appearance such as pose, scale, image rotation and orientation, illumination and facial expression make face detection a difficult pattern recognition problem. Hence, for face detection following problems need to be taken into account [5]:

1) Size: A face detector should be able to detect faces in different sizes. Thus, the scaling factor between the reference and the face image under test, needs to be given due consideration.
2) Expressions: The appearance of a face changes considerably for different facial expressions and thus, makes face detection more difficult.
3) Pose variation: Face images vary due to relative camera-face pose and some facial features such as an eye or the nose may become partially or wholly occluded. Another source of variation is the distance of the face from the camera, changes in which can result in perspective distortion.
4) Lighting and texture variation: Changes in the light source in particular can change a face's appearance can also cause a change in its apparent texture.
5) Presence or absence of structural components: Facial features such as beards, moustaches and glasses may or may not be present. And also there may be variability among these components including shape, colour and size.

The proposed system employs global feature matching for face recognition. However, all computation takes place only on the frontal face, by eliminating the hair and background as these may vary from one image to another. All systems therefore need frontal face extraction. One approach to achieve the aforementioned is by manually cropping the test image for required region or by precisely aligning the user's face with the camera before the test sample is clicked. Both these methods

may introduce a high degree of human error and so they have been avoided. Instead, automated Frontal Face Detection and Extraction is put to use. Therefore, a robust automatic face recognition system should be capable of handling the above problem with no need for human intervention. Thus, it is practical and advantageous to realize automatic face detection in a functional face recognition system. Commonly used methods for skin detection include Gabor filters, neural networks and template matching.

It has been proved that Gabor filters give optimum output for a wide range of variations in the test image with respect to user image, but it is the most time intensive procedure [8]. Moreover, it is unlikely that the test image would be severely out of sync for on the spot face recognition, so this method is not used. Most neural-network based algorithms [18],[19] are tedious and require training samples for different skin types which add to the already vast reference image database; hence, even this does not fit the program's requirements. Even template matching has severe drawbacks, including high computational cost [10] and fails to work as expected when the user's face is positioned at an angle in the test image.

After considering all the above factors, a classical appearance based methodology is applied to extract Frontal face. The default sRGB colour space is transformed to L*a*b* gamut, because L*a*b* separates intensity from *a* and *b* colour components [11]. L*a*b* colour is designed to approximate human vision in contrast to the RGB and CMYK colour models. It aspires to perceptual uniformity, and its *L* component closely matches human perception of lightness. It can thus be used to make accurate colour balance corrections by modifying output curves in the *a* and *b* components, or to adjust the lightness contrast using the *L* component. In RGB or CMYK spaces, which model the output of physical devices rather than human visual perception, these transformations can only be done with the help of appropriate blend modes in the editing application [10]. This distinction makes L*a*b* space more perceptually uniform as compared to sRGB and thus identifying tones, and not just a single colour, can be accomplished using L*a*b* space.

Fig. 1 shows the RGB colour model (B) relating to the CMYK model (C). The larger gamut of L*a*b* (A) gives more of a spectrum to work with, thus making the gamut of the device the only limitation.

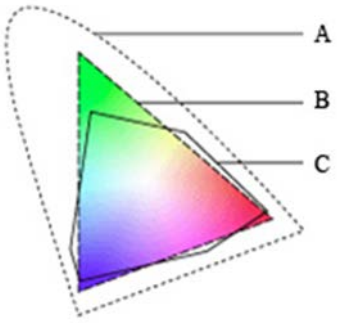


Fig. 1. RGB, CMYK and L*a*b* Colour Model

Tone identification is applied to detect skin by calculating the gray threshold of *a* and *b* colour components and then converting the image to pure black and white (BW) using the obtained threshold. Thus, RGB colour space can separate out only specific pigments, but L*a*b* space can separate out tones. Fig. 2 shows skin color differentiation in the form of white color using L*a*b* space whereas Fig. 4 shows no such differentiation using RGB. The extracted frontal face has been shown in Fig. 3.

Fig. 2. Reference image (left), image in black and white *a* plane (middle) and image in black and white *b* plane (right)

Fig. 3. Image in L*a*b* color space (left), frontal face extracted image (middle) and grayscale resized image (right)

Fig. 4. B&W image of red color space (left), B&W image of green color space (middle) and B&W image of blue color space (right)

There is higher probability of skin surface being the lighter part of the image as compared to the gray threshold [12], This may happen due to illumination and natural skin colour (in most cultures), so, pure white regions in the black and white image correspond to skin. It is assumed that face will have at least one hole, i.e., a small patch of absolute black due to eyes, chin, dimples etc. [11] and on the basis of presence of holes frontal face is separated from other skin surfaces like hands. A bounding box is created around the Frontal Face and after cropping the excess area, frontal face extraction is complete.

The above technique has been tested extensively on images obtained from standalone VGA cameras, webcams and camera equipped mobile devices having a resolution of 640 × 480. Even for resolutions as low as 320 × 200, where the test image is poorly illuminated or extremely grainy, the algorithm was able to successfully extract frontal face from test images. Thus, the proposed system is robust enough to achieve desired result even when low cost equipment like CCTV's and low resolution webcams are used. Also, since equipment with inferior picture quality like CCTV's and low resolution webcams are used, the

algorithm works as expected and hence can be called a low-cost approach to face detection. The extracted frontal face image is then fed as an input to the DWT-based face recognition process.

B. Normalization

Since the facial images are captured at different instants of the day or on different days, the intensity for each image may exhibit variations. To avoid these light intensity variations, the test images are normalized so as to have an average intensity value with respect to the registered image. The average intensity value of the registered images is calculated as summation of all pixel values divided by the total number of pixels. Similarly, average intensity value of the test image is calculated. The normalization value is calculated as:

$$\text{Normalization Value} = \frac{\text{Average value of reference image}}{\text{Average value of test image}} \quad (1)$$

This value is multiplied with each pixel of the test image. Thus we get a normalized image having an average intensity with respect to that of the registered image. Fig. 5 shows the test image and the corresponding normalized image.

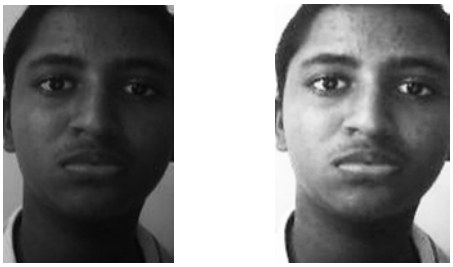


Fig. 5. Test Image and Normalized Image

C. Discrete Wavelet Transform

DWT [13] is a transform which provides the time-frequency representation. Often a particular spectral component occurring at any instant is of particular interest [14]. In these cases it may be very beneficial to know the time intervals these particular spectral components occur. For example, in EEGs, the latency of an event-related potential is of particular interest. DWT is capable of providing the time and frequency information simultaneously, hence giving a time-frequency representation of the signal. In numerical analysis and functional analysis, DWT is any wavelet transform for which the wavelets are discretely sampled. In DWT, an image can be analyzed by passing it through an analysis filter bank followed by decimation operation. The analysis filter consists of a low pass and high pass filter at each decomposition stage. When the signal passes through filters, it splits into two bands. The low pass filter which corresponds to an averaging operation, extracts the coarse information of the signal. The high pass filter which corresponds to a differencing operation, extracts the detail information of the signal. Fig. 6 shows the filtering operation of DWT.

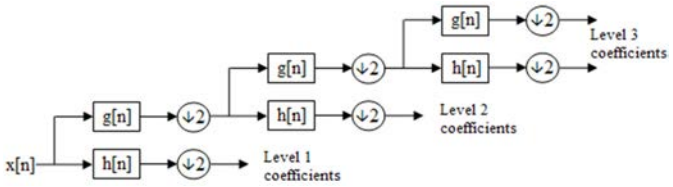
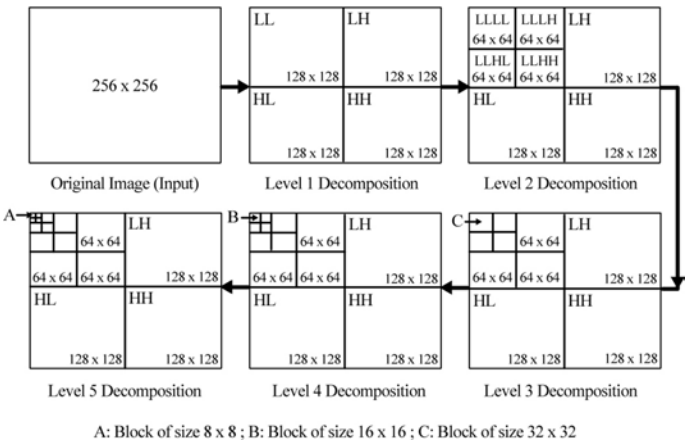


Fig. 6. DWT filtering operation

A two dimensional transform is accomplished by performing two separate one dimensional transforms. First the image is filtered along the row and decimated by two. It is then followed by filtering the sub image along the column and decimated by two. This operation splits the image into four bands namely LL, LH, HL and HH respectively. Further decompositions can be achieved by acting upon the LL sub band successively and the resultant image is split into multiple bands. For representational purpose, Level 2 decomposition of the normalized test image Fig. 5, is shown in Fig. 7.

Fig. 7. Level 2 DWT decomposition

At each level in the above diagram, the frontal face image is decomposed into low and high frequencies. Due to the decomposition process, the input signal must be a multiple of 2n where n is the number of levels. The size of the input image at different levels of decomposition is illustrated in Fig. 8.



A: Block of size 8 x 8 ; B: Block of size 16 x 16 ; C: Block of size 32 x 32

Fig. 8. Size of the image at different levels of DWT decomposition

The first DWT was invented by the Hungarian mathematician Alfréd Haar. The Haar wavelet [15] is the first

known wavelet and was proposed in 1909 by Alfred Haar. The term wavelet was coined much later. The Haar wavelet is also the simplest possible wavelet. Wavelets are mathematical functions developed for the purpose of sorting data by frequency. Translated data can then be sorted at a resolution which matches its scale. Studying data at different levels allows for the development of a more complete picture. Both small features and large features are discernable because they are studied separately. Unlike the Discrete Cosine Transform (DCT), the wavelet transform is not Fourier-based and hence, does a better job of handling discontinuities in data [16].

For an input represented by a list of 2n numbers, the Haar wavelet transform may be considered to simply pair up input values, storing the difference and passing the sum. This process is repeated recursively, pairing up the sums to provide the next scale, finally resulting in 2n − 1 differences and one final sum. Each step in the forward Haar transform calculates a set of wavelet coefficients and a set of averages. If a data set s_0, s_1, s_{N-1} contains N elements; there will be N/2 averages and N/2 coefficient values. The averages are stored in the lower half of the N element array and the coefficients are stored in the upper half. The averages become the input for the next step in the wavelet calculation, where for iteration i+1, $N_{i+1} = N_i/2$. The Haar wavelet operates on data by calculating the sums and differences of adjacent elements. The Haar equations to calculate an average (a_i) and a wavelet coefficient (c_i) from an odd and even element in the data set can be given as:

$$a_i = \frac{(S_i + S_{i+1})}{2} \tag{2}$$

$$c_i = \frac{(S_i - S_{i+1})}{2} \tag{3}$$

In wavelet terminology, the Haar average is calculated by the scaling function while the coefficient is calculated by the wavelet function.

D. Inverse Discrete Wavelet Transform

The data input to the forward transform can be perfectly reconstructed using the following equations:

$$S_i = a_i + c_i \tag{4}$$

$$S_i = a_i - c_i \tag{5}$$

After applying DWT, we take approximate coefficients, i.e., output coefficients of low pass filters. High pass coefficients are discarded since they provide detail information which serves no practical use for our application. Various levels of DWT are used to reduce the number of coefficients.

III. Implementation Steps

The image size used in the project work is 128 × 128 pixels. On applying the wavelet transform, the image is divided into approximate coefficients and detail coefficients. Level 1 yields the number of approximate coefficients as 64 × 64 = 4096. The approximate coefficients (low frequency coefficients) are stored and the detail coefficients (high frequency coefficients) are discarded. These approximate coefficients are used as inputs to the next level. Level 2 yields the number of approximate coefficients as 32 × 32 = 1024. These steps are repeated until an improvement in the recognition rate is observed. At each level the detail coefficients are neglected and the approximate coefficients are used as inputs to the next level. These approximate coefficients of input image and registered image are extracted. Each set coefficients belonging to the test image is compared with those of the registered image by taking the Euclidean distance and the recognition rate is calculated. Table 1 shows the comparison carried out at each level and its recognition rate.

TABLE I
Comparison of Various Levels of DWT

Levels	Coefficients	Recognition Rate without normalized image	Recognition Rate with normalized image
Level 1	4096	85.1%	91.4%
Level 2	1024	91.4%	91.4%
Level 3	256	93.6%	95.7%
Level 4	64	89%	93.6%
Level 5	16	87.6%	93.6%

Upon inspecting the results obtained, we can infer that Level 3 offers better performance in comparison to other levels. Hence the images are subjected to decomposition only up to Level 3.

IV. Future Work

The proposed face detection algorithm is time-efficient, i.e., having an execution speed of less than 1.75 seconds on an Intel Core 2 Duo 2.2 GHz processor. Due to its speed and robustness, it can further be extended for real time face detection and identification in video systems. Also, the proposed face recognition method can be coupled with recognition using local features, thus leading to an improvement in accuracy.

V. Conclusion

The proposed face detection and extraction scheme was able to successfully extract the frontal face for poor resolutions as low as 320 × 200, even when the original image was poorly illuminated or extremely grainy. Thus, images obtained from low cost equipment like CCTV's and low resolution webcams could be processed by the algorithm.

For face recognition, the recognition rate for global features using various levels of DWT was calculated. Generally, the recognition rate was found to improve upon normalization. Level 3 DWT Decomposition gives a superior recognition rate as compared to other decomposition levels.

References

[1] Z. Hafed, "Face Recognition Using DCT", *International Journal of Computer Vision*, 2001, pp. 167-188.

[2] W. Zhao, R. Chellappa, "Face Recognition: A Literature Survey", *ACM Computing Surveys*, Vol.35, No.4, December 2003, pp. 399-458, p. 9.

[3] S. Baker, T. Kanade, "Limits on super-resolution and how to break them," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 9, pp. 1167-1183, September 2002.

[4] A. Braun, I. Jarudi and P. Sinha, "Face Recognition as a Function of Image Resolution and Viewing Distance," *Journal of Vision*, September 23, 2011, Vol. 11 No. 11, Article 666.

[5] L. S. Sayana, M. Tech Dissertation, *Face Detection*, Indian Institute of Technology (IIT) Bombay, p. 5, pp. 10-15.

[6] C. Schneider, N. Esau, L. Kleinjohann, B. Kleinjohann, "Feature based Face Localization and Recognition on Mobile Devices," *Intl. Conf. on Control, Automation, Robotics and Vision*, Dec. 2006, pp. 1-6.

[7] L. V. Praseeda, S. Kumar, D. S. Vidyadharan, "Face detection and localization of facial features in still and video images", *IEEE Intl. Conf. on Emerging Trends in Engineering and Technology*, 2008,pp.1-2.

[8] K. Chung, S. C. Kee, S. R. Kim, "Face Recognition Using Principal Component Analysis of Gabor Filter Responses," *International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, 1999, p. 53.

[9] T. Kawanishi, T. Kurozumi, K. Kashino, S. Takagi, "A Fast Template Matching Algorithm with Adaptive Skipping Using Inner-Subtemplates' Distances," Vol. 3, *17th International Conference on Pattern Recognition*, 2004, pp. 654-65.

[10] D. Margulis, *Photoshop Lab Colour: The Canyon Conundrum and Other Adventures in the Most Powerful Colourspace*, Pearson Education. ISBN 0321356780, 2006.

[11] J. Cai, A. Goshtasby, and C. Yu, "Detecting human faces in colour images," *Image and Vision Computing*, Vol. 18, No. 1, 1999, pp. 63-75.

[12] Singh, D. Garg, *Soft computing*, Allied Publishers, 2005, p. 222.

[13] S. Jayaraman, S. Esakkirajan, T. Veerakumar, *Digital Image Processing*, Mc Graw Hill, 2008.

[14] S. Assegie, M.S. thesis, Department of Electrical and Computer Engineering, Purdue University, *Efficient and Secure Image and Video Processing and Transmission in Wireless Sensor Networks*, pp. 5-7.

[15] P. Goyal, "NUC algorithm by calculating the corresponding statistics of the decomposed signal" , *International Journal on Computer Science and Technology (IJCST)*, Vol. 1, Issue 2, pp. 1-2, December 2010 .

[16] Y. Ma, C. Liu, H. Sun, "A Simple Transform Method in the Field of Image Processing", *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications*, 2006, pp. 1-2.

[17] H. A. Rowley, S. Baluja and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, pp. 23-38, January 1998.

[18] P. Latha, L. Ganesan and S. Annadurai, "Face Recognition Using Neural Networks," *Signal Processing: An International Journal (SPIJ)*, Volume: 3 Issue: 5, pp. 153-160.

[19] S. Park, M. Park and M. Kang, "Super-resolution image reconstruction: a technical overview," *Signal Processing Magazine*, pp. 21-36, May 2003.

[20] P. Hennings-Yeomans, S. Baker, and B.V.K. Vijaya Kumar, "Recognition of Low-Resolution Faces Using Multiple Still Images and Multiple Cameras," *Proceedings of the IEEE International Conference on Biometrics: Theory, Systems, and Applications*, pp. 1-6, September 2008.

[21] MIT-CBCL Face Recognition Database, Center for Biological & Computational Learning (CBCL), Massachusetts Institute of Technology, Available: http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html, July 2011.

[22] Multi-PIE Database, Carnegie Mellon University, Available: http://www.multipie.org, July 2011.

[23] The Yale Face Database, Department of Computer Science, Yale University, Available: http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html, June 2011.

[24] T. Frajka, K. Zeger, "Downsampling dependent upsampling of images," *Signal Processing: Image Communication*, Vol. 19, No. 3, pp. 257-265, March 2004.

Divya P. Jyoti (M'2008) was born in Bhopal (M.P.) in India on April 24, 1990. She is currently pursuing her undergraduate studies in the Electronics and Telecommunication Engineering discipline at Thadomal Shahani Engineering College, Mumbai. Her fields of interest include Image Processing, and Human-Computer Interaction. She has 4 papers in International Conferences and Journals to her credit.

Aman R. Chadha (M'2008) was born in Mumbai (M.H.) in India on November 22, 1990. He is currently pursuing his undergraduate studies in the Electronics and Telecommunication Engineering discipline at Thadomal Shahani Engineering College, Mumbai. His special fields of interest include Image Processing, Computer Vision (particularly, Pattern Recognition) and Embedded Systems. He has 4 papers in International Conferences and Journals to his credit.

Pallavi P. Vaidya (M'2006) was born in Mumbai (M.H.) in India on March 18, 1985. She graduated with a B.E. in Electronics & Telecommunication Engineering from Maharashtra Institute of Technology (M.I.T.), Pune in 2006, and completed her post-graduation (M.E.) in Electronics & Telecommunication Engineering from Thadomal Shahani Engineering College (TSEC), Mumbai University in 2008. She is currently working as a Senior Engineer at a premier shipyard construction firm. Her special fields of interest include Image Processing and Biometrics.

M. Mani Roja (M'1990) was born in Tirunelveli (T.N.) in India on June 19, 1969. She has received B.E. in Electronics & Communication Engineering from GCE Tirunelveli, Madurai Kamraj University in 1990, and M.E. in Electronics from Mumbai University in 2002. Her employment experience includes 21 years as an educationist at Thadomal Shahani Engineering College (TSEC), Mumbai University. She holds the post of an Associate Professor in TSEC. Her special fields of interest include Image Processing and Data Encryption. She has over 20 papers in National / International Conferences and Journals to her credit. She is a member of IETE, ISTE, IACSIT and ACM.

Text-Independent Speaker Recognition for Low SNR Environments with Encryption

Aman Chadha
Department of Electronics & Telecommunication
TSEC
Mumbai, India
aman.x64@gmail.com

Divya Jyoti
Department of Electronics & Telecommunication
TSEC
Mumbai, India
dj.rajdev@gmail.com

M. Mani Roja
Department of Electronics & Telecommunication
TSEC
Mumbai, India
maniroja@yahoo.com

ABSTRACT
Recognition systems are commonly designed to authenticate users at the access control levels of a system. A number of voice recognition methods have been developed using a pitch estimation process which are very vulnerable in low Signal to Noise Ratio (SNR) environments thus, these programs fail to provide the desired level of accuracy and robustness. Also, most text independent speaker recognition programs are incapable of coping with unauthorized attempts to gain access by tampering with the samples or reference database. The proposed text-independent voice recognition system makes use of multilevel cryptography to preserve data integrity while in transit or storage. Encryption and decryption follow a transform based approach layered with pseudorandom noise addition whereas for pitch detection, a modified version of the autocorrelation pitch extraction algorithm is used. The experimental results show that the proposed algorithm can decrypt the signal under test with exponentially reducing Mean Square Error over an increasing range of SNR. Further, it outperforms the conventional algorithms in actual identification tasks even in noisy environments. The recognition rate thus obtained using the proposed method is compared with other conventional methods used for speaker identification.

General Terms
Biometrics, Pattern Recognition, Security

Keywords
Speaker Individuality, Text-independence, Pitch Extraction, Voice Recognition, Autocorrelation

1. INTRODUCTION
Humans have used body characteristics such as face, voice, gait, etc. for thousands of years to recognize each other. Alphonse Bertillon, chief of the criminal identification division of the police department in Paris, developed and then practiced the idea of using a number of body measurements to identify criminals in the mid-19th century. A wide variety of systems require reliable personal recognition schemes to either confirm or determine the identity of an individual requesting their services. The purpose of such schemes is to ensure that the rendered services are accessed only by a legitimate user and thus, disallow unauthorized access [1]. A biometric system is essentially a pattern recognition system that operates by acquiring biometric data from an individual, extracting a feature set from the acquired data, and comparing this feature set against the template set in the database. Thus, biometric systems fall under the ambit of technology designed and used specifically for measuring and analyzing the unique characteristics of a person. Any physiological and/or behavioral characteristic of a person can be used as a biometric feature as long as the following criteria are taken into account [2]:

1) Universality: Each person should have a characteristic which is distinct to the person in question;

2) Distinctiveness: Any two people should be sufficiently different in terms of their characteristics;

3) Permanence: The characteristic should be sufficiently invariant over a reasonable period of time;

4) Collectability: Measuring the characteristic quantitatively should be possible;

5) Performance: This refers to the achievable recognition accuracy and speed, the resources required to achieve the desired performance, as well as the operational and environmental factors that affect performance;

6) Acceptability: It indicates the extent to which people are willing to accept the use of a particular biometric identifier, i.e., the characteristic, in their daily lives;

7) Circumvention: This reflects how easily the system can be bypassed using fraudulent methods.

Even though reliable methods of biometric personal identification like finger-print analysis and retinal or iris scan do exist, however, the validity of forensic fingerprint evidence has recently been challenged by academics, judges and the media. While fingerprint identification was an improvement over earlier systems, the subjective nature of matching, along with the relatively high error rate of has made this forensic practice controversial. As far as iris recognition is concerned, it is very difficult to perform at a distance larger than a few meters and depends on the cooperation of the person. The initial investment in setting these systems is relatively high [2]. In contrast, voice recognition systems have the following advantages over other biometric identification systems:

- Users can enroll themselves over the telephone rather than having them enroll in person to deliver a fingerprint or an iris scan.
- The technology also requires no special data-acquisition system, other than a microphone. In case of signature verification systems, a specialized digital pen tablet acts as the data acquisition system whereas in iris

recognition systems, an Iris reader is deployed. Thus, hardware costs are reduced to a minimum.

- The voiceprint generated upon enrolment is characterized by the vocal tract, which is a unique physiological trait. A cold does not affect the vocal tract, so there will be no adverse effect on accuracy levels. Only extreme conditions such as laryngitis can hinder the optimal performance of the system.
- Voice recognition offers relatively low perceived invasiveness as compared to iris recognition, face recognition and signature verification.

Speaker recognition is the process of validating a user's claimed identity using characteristics extracted from their voices. No two individuals sound identical because their vocal tract shapes, larynx sizes and other parts of their voice production organs are different. In addition to these physical differences, each speaker has a distinctive manner of speaking, like the use of a particular accent, rhythm, intonation style, pronunciation pattern, choice of vocabulary etc. [3]. Depending on the context of the application, speaker recognition systems may operate either in verification mode or identification mode.

Speaker identification can be further divided into two branches: open-set identification and closed-set identification. Since it is generally assumed that imposters, i.e., those assuming identity of valid users, are not known to the system, this is referred to as an open-set task. Generally it is assumed the unknown voice must come from a fixed set of known speakers, thus the task is often referred to as closed-set identification. In this paper, we deal with an instance of closed-set speaker identification.

Depending on the algorithm used for the identification, the process can be categorized as text-dependent or text-independent identification. If the text must be the same for enrollment and verification this is called text-dependent recognition. In text-dependent systems, suited for cooperative users, the recognition phrases are known beforehand. For instance, the user can be prompted to read a randomly selected sequence of numbers as illustrated in [7]. In text-independent systems, there are no constraints on the words which the speakers are allowed to use. Thus, the reference and the test utterances may have completely different content. Text-independent systems are most often used for speaker identification as they require very little, if any, cooperation by the speaker. In fact, enrollment may happen without the user's knowledge, as in the case for many forensic applications [3].

Most Biometric Recognition systems are used as security gateways which control access to sensitive information. In such programs if a false negative is triggered, it can be corrected in most cases, by testing again, whereas a false positive may have disastrous consequences from the view point of Data Security and Integrity. Thus, the accuracy rate must be calculated by taking into account both- samples that the program failed to recognize and samples which were identified incorrectly. Further, in text independent speaker recognition systems, an imposter may gain permissions by the following means:

1) A mimicry artist may be employed to imitate the speaker's voice and diction.

2) The speaker's voice may be recorded without his or her consent and knowledge, and this sample may be played to the testing software.

3) The imposter may replace the reference sample of the user in the database by his or her own voice sample.

4) Easily available voice changer software may be used by an imposter to mimic the voice of any reference sample if the database is vulnerable.

All the above scenarios will result in a false positive result due to short comings of a Recognition system based purely on voice. Thus, the paper proposes extensive use of Encryption by means of a private-key which generated from the password selected by the user. This key is then used to seed two levels of Pseudo Random Noise Generators (PRNG) for scrambling the signal sandwiched with Transform-based encryption to increase the robustness of encoding algorithm.

The performance of automatic speech recognizers (ASR) has known to degrade rapidly in the presence of noise and other distortions [6]. Speech recognizers are typically trained on clean speech and typically render inferior performance when used in conditions where speech occurs simultaneously with other unwarranted sound sources, i.e., distortions and disturbances. Some of these unsolicited sources are as below:

- Speech recorded with a microphone or telephone handset is generally vulnerable to environmental noise such as computer hum, car engine, door slams, echoing, keyboard clicks, traffic noise, background noise, which adds to the speech wave [10].
- Reverberation adds delayed versions of the original signal to the recorded signal [4].
- The A/D converter adds its own distortion, and the recording device might interfere with mobile phone radio-waves.
- If the speech is transmitted through a telephone network, it is compressed using lossy techniques which might have added noise into the signal.

To sum up, the speech wave fed to the recognition algorithm is not the same wave that was transmitted from the speaker's lips and nostrils, but it has gone through several transformations degrading its quality [10]. If samples of the corrupting noise source are available before hand, a model for the noise source can additionally be trained and noisy speech may be jointly decoded using trained models of speech and noise [9]. However, in many realistic applications, adequate amounts of noise samples are unavailable before-hand, and hence training of a noise model is not feasible. Fig. 1 illustrates some additive and convolutive noise sources which occur during the process of speaker recognition.

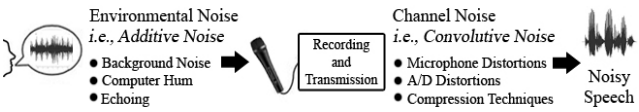


Fig 1: Error sources during the process of speaker recognition

The Mel-Frequency Cepstral Coefficients (MFCC) are the most evident example of a feature set that is extensively used in speaker recognition. When MFCC front-end is used in speaker recognition system, one makes an implicit assumption that the human hearing mechanism is the optimal speaker recognizer. However, this has not been confirmed, and in fact opposite results exist [10]. In most speaker recognition systems, MFCC

has shown to achieve fairly good performance. Conventionally, MFCC features are extracted from the spectral analysis of 20 to 30 ms long speech frames with an overlap of 10 to 15 ms [17],[18]. The length of the analysis window and the size of overlap are usually fixed for each system. The drawbacks of a fixed length analysis window have been issued by many researchers [18],[23]. This paper presents a modified version of the autocorrelation pitch extraction algorithm robust against noise.

2. IDEA OF THE PROPOSED SOLUTION

The primary objective of this paper is to implement a speaker recognition system sustainable to a great extent, against noise, i.e., a system offering superior performance under low SNR conditions. Moreover, for the system to offer high levels of security, a robust multi-level encryption scheme needs to be implemented. The database used for this project consists of voice samples of 50 subjects. 6 voice samples are taken for each person. Out of these, 3 are used for training and the remaining samples are used for testing. Test voice samples with different tones and volume levels are considered for the experiment.

2.1 Need for Reference Template Encryption

Encryption is the process of transforming data into scrambled unintelligible cipher text using a key. The role of Encryption is to secure information when it is stored or in transit. However, it is relatively easier to crack single level encryption by brute force or correlation, as compared to a multi-level encryption scheme. Therefore, the program requires a user defined 8-character password to seed two out of three levels of encryption. The password must have a minimum of one capital alphabet, one numeral and one special character, such as @, >, & etc. Fig. 2 shows how, more than 6.6×10^{15} permutations are possible. This password is processed further using numeric substitutions in Caesar's Cipher type of encryption and a state seed is generated. Findings in [5] show that a hacker may take as less as 10 minutes to crack each password once a rainbow table has been built, if all passwords are stored internally in a memory hash. Hence, the system is designed to store no password and the state seed generated on the fly is divided into two keys, each of which is used for further encryption and computation.

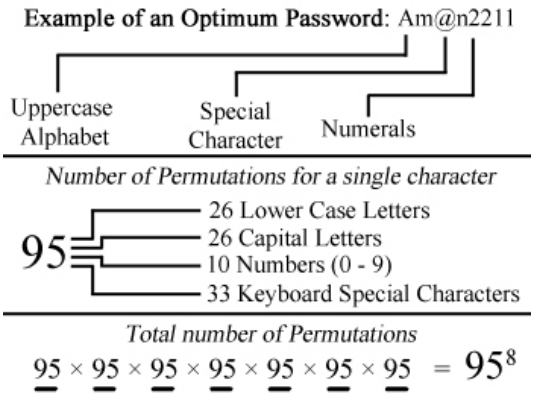


Fig 2: Permutations for a password

Random numbers can be generated by a random bit generator which can be defined as a device or algorithm whose output is a sequence of statistically independent and unbiased binary digits. Pseudorandom Number Generator (PRNG) [12] is used to generate random bits dependent on the state seed, such that an adversary cannot judge the next bit by correlating a subset of the random bits generated. This is ensured by virtue of large number of internal states of the generator which in turn means large period of the random bits generated. For implementing the proposed encryption algorithm, PRNG inbuilt in MATLAB 7 is used which has an average period of 235×16 as total number of internal states are 35 words. In short, on an average the random sequence will repeat itself only after 235×16 bits, which is much greater than the length of the samples required for testing.

PRNG works by taking a state seed s as input and generating the output sequence of random values as f(s), f(s+1), f(s+2), … Here, f is a one way function which can be defined as an algorithm whose output is a sequence of statistically independent and unbiased binary digits [12]. Based on the properties of this function f, some output values say, f(s+i), must be discarded to eliminate correlation between subsequent random bits. This approach is actively followed by standardized one-way functions, for e.g., a cryptographic hash function such as SHA-1 or a block cipher such as DES (x7.4). When the sequence so generated is superimposed with the reference or test voice sample, scrambling takes place and signal is rendered nearly indecipherable. Yet, the signal is still in time domain and a person may make a correct guess of the scrambling key. Thus, the second type of encryption used is Transform based encryption.

A transform based encryption changes the domain of the data from say time domain to frequency domain or complex plot etc. It means that the new data which is acquired has no meaning in its previous domain, in this case-time domain. A discrete cosine transform (DCT) based scheme is used for further scrambling as [11] illustrates the superiority of DCT over four other discrete transform based encryption techniques for analog speech, when compared with respect to a novel cryptanalytic attack. DCT is a well-known transform that decomposes a signal into its frequency components and it un-correlates the sequence of input samples, i.e. DCT coefficients give the frequency domain equivalent of speech data [11]. In fact, for large databases this step may be used to compress the voice signal at the cost of system accuracy. A final layer of pseudorandom noise using the second part of the state key generated at run-time is superimposed on the noisy signal DCT coefficients and source side encryption is complete. The sample so obtained is amplified and assumed to be transmitted over an AWGN channel with known Signal to Noise Ratio (SNR). At the testing site, this received sample is decrypted in inverse order of encryption and matched with test sample for Speaker Recognition.

2.2 Speaker Recognition Process

Fig. 3 shows the general structure of the speaker recognition system.

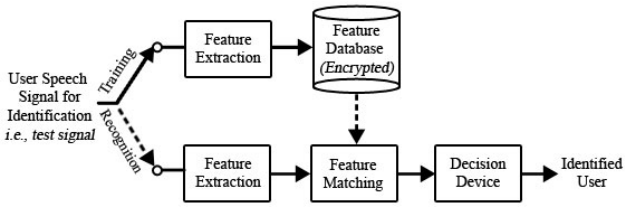


Fig 3: General architecture of a speaker recognition system

This system operates in two modes: training and recognition. In the training mode a new speaker (with a known identity) is enrolled into the database, while in the recognition mode an unknown speaker gives a speech input signal and the system tries to identify the speaker.

1) Feature Extraction: The feature extractor, i.e, the front-end, is the first component in an ASR system. Feature extraction transforms the raw speech signal into a compact but effective representation that is more stable and discriminative than the original signal.

2) Database: A collection of voice samples has been recorded for evaluation of the proposed system. The recordings were converted to WAV format to facilitate easy analysis and operations using MATLAB 7. The WAV files were subjected to a multi-level encryption scheme to foster maximum security.

3) Speaker Modeling: The training phase uses the acoustic vectors extracted from each segment of the signal to create a speaker model which will be stored in a database.

4) Pattern matching and decision: The Pattern matching strategy takes all the matching scores from the user pattern to each of the stored reference patterns into account and searches for the "closest" possible match and thus makes a decision.

The steps involved in the entire process can be summarized as follows:

1) The locations of the samples of different users are stored upon encryption them using the proposed multi-level encryption scheme.

2) Various features and parameters of the voice samples such as mean, variance, standard deviation, pitch etc. are calculated.

3) The voice sample of a test speaker is recorded.

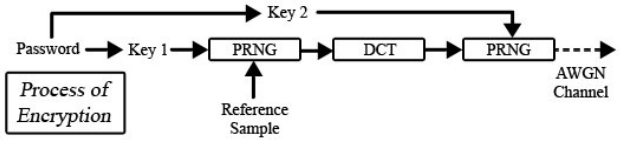
4) The Euclidean distance between the features of this sample and the samples previously stored in the database is calculated.

5) The Euclidean distances are then arranged in an ascending order, with the first Euclidean distance being the minimum.

6) The sample corresponding to the first Euclidean distance is the sample having the highest resemblance to the sample under test.

3. IMPLEMENTATION STEPS

Pitch, i.e., fundamental frequency, is an important parameter of speech signals which is used in speech analysis, synthesis and recognition. Fundamental frequency (F0) as an acoustic correlate is strongly related to prosodic information of stress and intonation. For speech recognition applications, pitch extraction (fundamental frequency estimation) provides the basis for voiced/unvoiced classification decision. However, before pitch extraction and subsequent matching takes place, the Reference voice sample must be encrypted, transmitted over AWGN channel and decrypted at testing site. Fig. 4 shows the data flow at various steps.



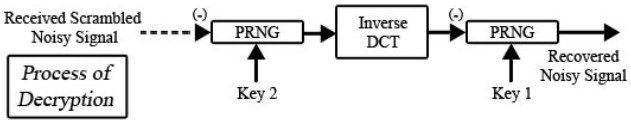


Fig 4: Scheme for encryption and decryption

The steps involved in the process of generation of the two State Keys can be summarized as follows:

1) Read an 8 character long password from user.

2) Convert to ASCII equivalent to form an array say 'x'.

3) Apply Caesar's cipher with a shift of 4, to each ASCII element to form a new array say 'y'.

4) Concatenate all elements of y to form a single integer say 'z'.

5) Calculate length of 'z'.

6) If even break into two equal halves two generate two equal length keys – Key1 and Key2.

7) Else break asymmetrically, the longer key is Key1 and shorted key is Key2.

Sample Password: Djyot!24

ASCII Array x: [68 106 121 64 116 105 50 52]

Cipher Text Array y: [72 110 125 69 120 109 54 56]

Concatenated Integer z: 72110125691201095456

Key1: 7211012569 **Key2**: 1201095456

Fig 5: An example illustrating generation of keys

The steps involved in the process of encryption can be summarized as follows:

1) Level 1: Key1 is passed as seed to PRNG and superimpose the output sequence on reference voice sample to generate a noisy signal say 'x'.

2) Level 2: Perform DCT on x and save the coefficients in an array say 'y'.

3) Level 3: Key2 is passed as seed to PRNG and superimpose the random sequence obtained on y and save the resultant as an array say 'z'.

4) z is the signal which is to be transmitted through AWGN channels with known SNR. Thus, it is subjected to various SNR levels and the resulting signal is normalized and then decrypted as follows.

The steps involved in the process of decryption can be summarized as follows:

1) Level 3: Key2 is passed as seed to PRNG and algebraically subtract the random sequence so obtained from the received signal, save the resultant as say 'x'.

2) Level 2: Take Inverse DCT of x and save the noisy signal so obtained as 'y'.

3) Level 1: Key1 is passed as seed to PRNG and algebraically subtract the random sequence so obtained from x, save the resultant as say, 'z'.

4) The file 'z', recovered after the above steps is the final decrypted version of reference signal, this is sent for pitch extraction along with the test signal and comparative matching takes place.

Fig. 6 and Fig. 7 show the encryption and decryption plots of the sample under test respectively.

Fig 6: Original signal (Left), signal after Level 2 (Middle) and signal after Level 2 and AWGN Insertion (Right)

Fig 7: Received signal (Left), recovered file with high SNR (Middle) and recovered file with low SNR (Right)

Pitch extraction, also known as, fundamental frequency estimation, plays a vital role in speech processing and has numerous applications in speech related areas. Therefore, several methods to extract the pitch of speech signals have been proposed by researchers. However, such methods are known to be very vulnerable in noisy environments, hence, performance improvement in noisy environments is still desired. For example, this is particularly true in speech enhancement systems, because in such systems the accuracy of pitch extraction is directly related with the quality of speech after the operations of enhancement. Also, speech communication systems often transmit pitch information. To do this, we have to extract the pitch of speech signals in practical noisy environments. Unfortunately, a reliable and accurate method for pitch extraction in noisy environments is still a subject of scientific investigation.

Generally, pitch detection algorithms (PDA) use short-term analysis techniques [16]. For every frame x_m we get a score $f(T|x_m)$ which is a function of the candidate pitch period T. Such algorithms, in general, offer a rough estimation of the pitch by maximizing the following equation:

$$T_m = \arg\max_T f(T \mid x_m) \quad (1)$$

A commonly used method to estimate pitch is based on detecting the highest value of the autocorrelation function in the region of interest. The correlation between two waveforms is a measure of their similarity. The waveforms are compared at different time intervals, and their similarity is calculated at each interval. The result of a correlation is a measure of similarity as a function of time lag between the beginnings of the two waveforms. One would expect exact similarity at a time lag of zero, with increasing dissimilarity as the time lag increases. The mathematical definition of the autocorrelation function $R_{xx}(\tau)$ is shown in (2), for an infinite discrete function $x[n]$, and (3) shows the mathematical definition of the autocorrelation $R_{xx}(\tau)$ of a finite discrete function $x'[n]$ of size N.

$$R_{xx}(\tau) = \sum_{n=-\infty}^{\infty} x[n]x[n+\tau] \quad (2)$$

$$R_{xx}'(\tau) = \sum_{n=0}^{N-1-\tau} x'[n]x'[n+\tau] \quad (3)$$

where, $x[n]$ is the speech signal;

τ is the lag number;

n is the time for a discrete signal.

Correlation based processing is known to be comparatively robust against noise and may be one which provides the best performance in noisy environments [19],[20],[22]. The autocorrelation function of a signal is actually a non-invertible transformation of the signal that is useful for displaying structure in the waveform. Hence for pitch detection applications, if we assume $x(n) = x(n + P)$ for all n, i.e., $x(n)$ is periodic with period P, then it is easily shown that:

$$R_{xx}(\tau) = R_{xx}(\tau + P) \quad (4)$$

Equation (4) basically indicates that the autocorrelation function is also periodic with the same period. Conversely, periodicity in the autocorrelation function indicates periodicity in the original signal.

Speech being a non-stationary signal, the concept of a long-time autocorrelation measurement as defined in (3) cannot be easily extrapolated for such signals [16]. Thus, it is mandatory to define a short-time autocorrelation function, which operates on short segments of the signal as:

$$R_{xx}(\tau) = \frac{1}{N}\sum_{n=0}^{N'-1}\left[x(n+l)w(n)\right]\left[x(n+l+\tau)w(n+\tau)\right] \quad (5)$$

where, $0 \le \tau \le M_0$;

$w(n)$ is an appropriate window for analysis;

N is the section length being analyzed;

N' is the number of signal samples used in the computation of $R_{xx}(\tau)$;

M_0 is the number of autocorrelation points to be computed;

τ is the lag number;

l is the index of the starting sample of the frame.

For applications involving pitch estimation, N' is generally set to the value given by the following equation:

$$N' = N - m \quad (6)$$

This is done so that only the N samples present in the analysis frame, i.e., $x(l)$, $x(l + 1)$, . . . , $x(l + N - 1)$ are used in the autocorrelation computation. Values of 200 and 300 have generally been used for M_0 and N respectively corresponding to a maximum pitch period of 20 ms (200 samples at a 10 kHz sampling rate) and a 30 ms analysis frame size.

Correlation based processing also includes the average magnitude difference function (AMDF) method [15],[16],[18]. The AMDF PDA is chosen in our study is because it has

relatively low computational cost and is easy to implement. Several types of noise such as babble, car, and street noises with 5 dB, 10 dB, 15 dB and 20 dB SNR are used to evaluate the performance of the proposed system. The noise sources are taken from the NOISEX-92 database [21], from Carnegie Mellon University, a collection of different noise waveforms which can be used to generate speech waveforms in various noise conditions and with different signal to noise ratio (SNR) values. The AMDF is [13] essentially a variation of autocorrelation function analysis where, instead of correlating the input speech at various delays (where multiplications and summations are formed at each value), a difference signal is formed between the delayed speech and the original, and at each delay value the absolute magnitude is taken [14]. In contrast with the autocorrelation or cross-correlation function, the AMDF calculations require no multiplications, a much sought-after property for real-time applications. Therefore, for the purpose of emphasizing the true peak produced by the autocorrelation, i.e., for measuring the periodicity of voiced speech, we propose an autocorrelation function (AMDF). The AMDF is defined by the following equation:

$$AMDF(\tau) = \frac{1}{N}\sum_{n=0}^{N-1}\left|x[n] - x[n-\tau]\right| \quad (7)$$

where, *x(n)* are the samples of input speech;

x(n - τ) are the samples obtained by introducing a delay of *τ* seconds.

Equation (7) indicates the characteristic of the AMDF that when x[n] is similar with x[n - τ], AMDF(τ) yields a small value. A difference signal is thus formed by delaying the input speech various amounts, subtracting the delayed waveform from the original and summing the magnitude of the differences between sample values. For zero delay, the difference signal is always zero and is particularly small at delays corresponding to the pitch period of a voiced sound having a quasi-periodic structure [16].

For each value of delay, computation is made over an integrating window of *N* samples. To generate the entire range of delays, the window is "cross differenced" with the full analysis interval. An advantage of this method is that the relative sizes of the nulls tend to remain constant as a function of delay, which is mainly because there is always full overlap of data between the two segments being cross differenced. In extractors of this type, the limiting factor on accuracy is the inability to completely separate the fine structure from the effects of the spectral envelope. For this reason, decision logic and prior knowledge of voicing are used along with the function itself to help make the pitch decision more reliable.

Let us assume that *x(n)* is a noisy speech signal composed of the actual speech content and the Additive White Gaussian Noise (AWGN). *x(n)* is given by the following equation:

$$x(n) = s(n) + w(n) \quad (8)$$

where, *s(n)* is a clean speech signal;

w(n) is the AWGN.

The autocorrelation function $R_{xx}(\tau)$ for this particular case, as demonstrated in [15], is given by:

$$\begin{aligned} &= \frac{1}{N}\sum_{n=0}^{N-1}(s[n] + w[n]) \cdot (s[n+\tau] + w[n+\tau]) \\ &= \frac{1}{N}\sum_{n=0}^{N-1}(s[n]s[n+\tau] + s[n]w[n+\tau] \\ &\quad + w[n]s[n+\tau] + w[n]w[n+\tau]) \\ &= R_{ss}(\tau) + 2R_{sw}(\tau) + R_{ww}(\tau) \end{aligned}$$

where, $R_{ss}(\tau)$ is the autocorrelation function of *s[n]*;

$R_{sw}(\tau)$ is the crosscorrelation function of *s[n]* and *w[n]*;

$R_{ww}(\tau)$ is the autocorrelation function of *w[n]*.

In [15], the case for large values of N has been described. If the speech signal shows no correlation with the AWGN, then $R_{sw}(\tau)$ does not exist, i.e., it yields a zero value. The following equation exists for this case:

$$R_{xx}(\tau) = R_{ss}(\tau) + R_{ww}(\tau) \text{ ...if } \tau = 0 \quad (8)$$

Also, if *w[n]* is uncorrelated, then $R_{ww}(\tau)$ yields a zero value except for $\tau = 0$. If this is the case, the following relation holds true:

$$R_{xx}(\tau) = R_{ss}(\tau) \text{ ...if } \tau \neq 0 \quad (9)$$

Based on the above mentioned properties, the autocorrelation function provides robust performance against noise. When the characteristics of the AMDF are plotted, it is found to yield a notch, while the autocorrelation function yields a peak. Thus, the characteristics of the AMDF are found to bear similarity with that of the autocorrelation function. However, both functions have the same periodicity. Pitch of the segmented speech is estimated by searching the peak of the resultant function obtained on coupling the autocorrelation function with the AMDF. However, upon deploying the resultant function directly, we observe that the accuracy of pitch extraction is compromised. Therefore, the system uses interpolation based on 3 points around the detected peak [15]. It is known that such interpolation on the autocorrelation function is useful for improving the accuracy of pitch extraction. Lagrange's method is used to perform the interpolation operation. The frequency band selected for searching the pitch peak is from 50 Hz to 400 Hz as it corresponds to the region of the fundamental frequencies of most men and women. The technique of removing the formant structure for reliable pitch detection by center clipping demonstrated by M. Sondhi [8] while still retaining the pitch period information was implemented to reduce the effects of the formant structure on the detailed shape of the short-time autocorrelation function.

4. RESULTS

Upon performing Decryption on the signals that were subjected to AWGN, the Mean Square Error (MSE) values thus obtained, are tabulated against SNR as follows:

Table 1. MSE values for corresponding SNR values

SNR in dB	Mean Square Error
16	2.83×10^{-2}
17	1.82×10^{-2}
18	2.02×10^{-3}
19	4.11×10^{-5}
20	7.32×10^{-7}

Fig. 8 shows the plot of the MSE values against the SNR values.

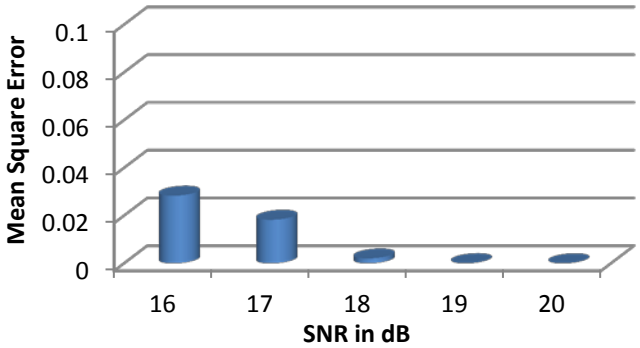


Fig 8: Plot of MSE against corresponding SNR

To investigate the accuracy of the modified pitch extraction method, various experiments were conducted to compare the efficiency of the algorithm with four standard conventional speaker identification methods namely, Statistical Methods (Mean, Moment and Variance), Linear Predictive Coding (LPC), Zero Crossing and Fast Fourier Transform (FFT).

Table 2. Results obtained using the proposed modified autocorrelation method and comparison with other methods

Algorithm	Accuracy
Pitch Extraction	92.39
Mean, Moment and Variance	74.87
Linear Predictive Coding	72.42
Zero Crossing	62.35
Fast Fourier Transform	55.49

Fig. 9 shows the plot of accuracy rates of various algorithms.

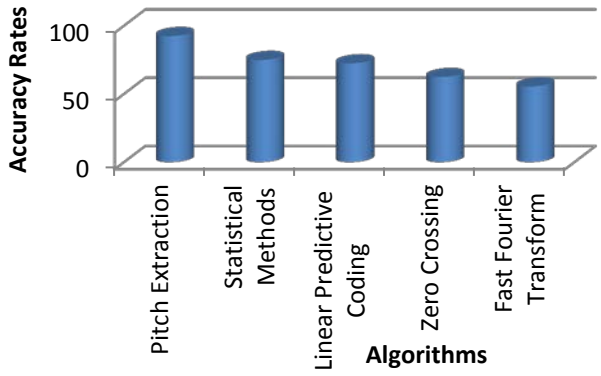


Fig 9: Plot of various algorithms against their accuracy rates

5. FUTURE SCOPE

The system can be implemented for real time applications if the database can be standardized online, this will eliminate the need of training samples for the system. Also, the concept of voice recognition using autocorrelation can be tried for a larger database. As the size of the database increases, encryption algorithm could be further strengthened by using longer keys and increasing the period of pseudorandom noise sequences which shall make decryption by brute force near impossible and also reduce MSE to a minimum.

6. CONCLUSION

The primary objective of the paper was to implement a robust and secure voice recognition system using minimum resources offering optimum performance in noisy environments. We have implemented this system using three levels of encryption for data security and autocorrelation based approach to find the pitch of the sample. The resulting system was found to reduce significantly the amount of test data or features to be extracted. By virtue of DCT based scrambling, the system is highly immune to cryptanalytic attacks that target the redundancy of speech. The robustness of the system in adverse conditions such as noisy or channel distorted environments was verified by conducting closed set text-independent speaker identification experiments, and results pointed to improved performance in adverse SNR environments. We conclude that the proposed algorithm is equipped with means to ensure that data security is not compromised at any stage of computation and at the same time high accuracy rate of the pitch detection algorithm makes it very powerful in both clean and noisy environments.

7. REFERENCES

[1] A. K. Jain, A. Ross, S. Prabhakar, "An introduction to biometric recognition," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No.1, Jan. 2004, 4- 20.

[2] R. Clarke, “Human Identification in Information Systems: Management Challenges and Public Policy Issues,” Information Technology & People, Vol. 7, No. 4, 6–37, 1994.

[3] T. Kinnunen, H. Lib, “An Overview of Text-Independent Speaker Recognition: from Features to Supervectors,” 1-3.

[4] X. Huang, A. Acero, and H. W. Hon, Spoken Language Processing: a Guide to Theory, Algorithm, and System Development, Prentice-Hall, New Jersey, 2001.

[5] K. Theocharoulis, I. Papaefstathiou, C. Manifavas, “Implementing Rainbow Tables in High-End FPGAs for Super-Fast Password Cracking,” International Conference on Field Programmable Logic and Applications, 2010, 145-150.

[6] X. Huang, A. Acero, and H. Hon, Spoken Lang. Process.. Upper Saddle River, NJ: Prentice-Hall, 2001.

[7] A. Higgins, L. Bahler, and J. Porter, “Speaker verification using randomized phrase prompting,” Digital Signal Processing 1 (April 1991), 89–106.

[8] M. Sondhi, “New methods of pitch extraction,” IEEE Transactions on Audio and Electroacoustics, Vol. 16, No. 2, Jun 1968, 262-266.

[9] A. P. Varga and R. K. Moore, “Hidden Markov model decomposition of speech and noise,” in Proc. ICASSP’90, 1990, 845-848.

[10] T. Kinnunen, Licentiate’s Thesis, “Spectral Features for Automatic Text-Independent Speaker Recognition,” Department of Computer Science, University of Joensuu, December 2003, 5-11, 2-3.

[11] B. Goldburg, S. Sridharan, E. Dawson, "Design and cryptanalysis of transform-based analog speech scramblers," IEEE Journal on Selected Areas in Communications, June 1993, 735-744.

[12] A. Menezes , P. Oorschot , S. Vanstone , R. Rivest, Handbook of Applied Cryptography, CRC Press, 1996, 169-179, 180-190.

[13] M. J. Ross, H. L. Shaffer, A. Cohen, R. Freudberg, and H. J. Manley, "Average magnitude difference function pitch extractor," IEEE Transactions on Acoustics, Speech, Signal Processing, Vol. ASSP-22, Oct. 1974, 353-362.

[14] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg, and C. A. McGonegal, "A comparative performance study of several pitch detection algorithms," IEEE Transactions on Audio, Signal, and Speech Processing 24, 1976, 399-417.

[15] H. Kobayashi and T. Shimamura, "A weighted autocorrelation method for pitch extraction of noisy speech," IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 3, 2000, pp.1307-1310.

[16] L. Tan and M. Karnjanadecha, "Pitch Detection Algorithm: Autocorrelation and AMDF", International Symposium on Communications and Information Technologies (ISCIT 2003), 1-4.

[17] L. R. Rabiner, B.H. Juang, Fundamentals of speech recognition, Prentice Hall, 1993.

[18] T. F. Quatieri, Discrete time speech signal processing, Prentice Hall, 2002.

[19] S. Jungpa, S. Hong, J. Gu, M. Kim, I. Baek, Y. Kwon, K. Lee, Sung-I Yang, "New speaker recognition feature using correlation dimension," Proceedings of the IEEE International Symposium on Industrial Electronics, 2001, ISIE 2001, Vol.1, 2001, pp.505-507.

[20] S. Kim, M. Ji, H. Kim, "Robust speaker recognition based on filtering in autocorrelation domain and sub-band feature recombination," Pattern Recognition Letters, Elsevier Science, Vol. 31, No. 7, May 2010, 593-599.

[21] NOISEX-92 Database, Carnegie Mellon University, http://www.speech.cs.cmu.edu/comp.speech/Section1/Data/noisex.html, July 2011.

[22] L. R. Rabiner, "On the use of autocorrelation analysis for pitch detection," IEEE Transactions on Acoustics, Speech, Signal Processing, Vol. ASSP-25, No. 1, 24–33, Feb. 1977.

[23] Y. J. Kim, and J. H. Chung, "Pitch synchronous cepstrum for robust speaker recognition over telephone channels," Electroincs letters, Vol. 40, No. 3, 207–209, 2004.

AUTHOR BIOGRAPHIES

Aman Chadha (M'2008) was born in Mumbai (M.H.) in India on November 22, 1990. He is currently pursuing his undergraduate studies in the Electronics and Telecommunication Engineering discipline at Thadomal Shahani Engineering College, Mumbai. His special fields of interest include Image Processing, Computer Vision (particularly, Pattern Recognition) and Embedded Systems. He has 5 papers in International Conferences and Journals to his credit. He is a member of IETE, IACSIT and ISTE.

Divya Jyoti (M'2008) was born in Bhopal (M.P.) in India on April 24, 1990. She is currently pursuing her undergraduate studies in the Electronics and Telecommunication Engineering discipline at Thadomal Shahani Engineering College, Mumbai. Her fields of interest include Image Processing, and Human-Computer Interaction. She has 4 papers in International Conferences and Journals to her credit.

M. Mani Roja (M'1990) was born in Tirunelveli (T.N.) in India on June 19, 1969. She has received B.E. in Electronics & Communication Engineering from GCE Tirunelveli, Madurai Kamraj University in 1990, and M.E. in Electronics from Mumbai University in 2002. Her employment experience includes 21 years as an educationist at Thadomal Shahani Engineering College (TSEC), Mumbai University. She holds the post of an Associate Professor in TSEC. Her special fields of interest include Image Processing and Data Encryption. She has over 20 papers in National / International Conferences and Journals to her credit. She is a member of IETE, ISTE, IACSIT and ACM.

Proceedings of the 2011 International Conference on Recent Advancements in Electrical, Electronics and Control Engineering (IConRAEeCE)
IEEE Xplore: CFP1153R-ART; ISBN: 978-1-4577-2149-6

Face Recognition Using Discrete Cosine Transform for Global and Local Features

Aman R. Chadha, Pallavi P. Vaidya, M. Mani Roja
Department of Electronics & Telecommunication
Thadomal Shahani Engineering College
Mumbai, INDIA
aman.x64@gmail.com

***Abstract*— Face Recognition using Discrete Cosine Transform (DCT) for Local and Global Features involves recognizing the corresponding face image from the database. The face image obtained from the user is cropped such that only the frontal face image is extracted, eliminating the background. The image is restricted to a size of 128 × 128 pixels. All images in the database are gray level images. DCT is applied to the entire image. This gives DCT coefficients, which are global features. Local features such as eyes, nose and mouth are also extracted and DCT is applied to these features. Depending upon the recognition rate obtained for each feature, they are given weightage and then combined. Both local and global features are used for comparison. By comparing the ranks for global and local features, the false acceptance rate for DCT can be minimized.**

Keywords- face recognition; biometrics; person identification; authentication; discrete cosine transform; DCT; global local features.

I. INTRODUCTION

A face recognition system is essentially an application [1] intended to identify or verify a person either from a digital image or a video frame obtained from a video source. Although other reliable methods of biometric personal identification exist, for e.g., fingerprint analysis or iris scans, these methods inherently rely on the cooperation of the participants, whereas a personal identification system based on analysis of frontal or profile images of the face is often effective without the participant's cooperation or intervention. One of the many ways for automatic identification or verification is by comparing selected facial features from the image and a facial database. This technique is typically used in security systems. For e.g., the technology could be used as a security measure at ATM's; instead of using a bank card or personal identification number, the ATM would capture an image of the person's face, and compare it to his/her photo in the bank database to confirm the identity of the relevant person. On similar lines, this concept could also be extrapolated to computers; by using a web cam to capture a digital image of a person, the face could replace the commonly used password as a means to log-in and thus, authenticate oneself. Given a large database of images and a photograph, the problem is to select from the database a small set of records such that one of the image records matched the photograph. The success of the method could be measured in terms of the ratio of the answer list to the number of records in the database. The recognition problem is made difficult by the great variability in head rotation and tilt, lighting intensity and angle, facial expression, aging, etc. Some other attempts at facial recognition by machines have allowed for little or no variability in these quantities. A general statement of the problem of machine recognition of faces can [2] be formulated as: given a still or video image of a scene, identify or verify one or more persons in the scene using a stored database of faces. Available collateral information such as race, age, gender, facial expression, or speech may be used in narrowing the search. The solution to the problem involves segmentation of faces, feature extraction from face regions, recognition, or verification. In identification problems, the input to the system is an unknown face, and the system reports back the determined identity from a database of known individuals, whereas in verification problems, the system needs to confirm or reject the claimed identity of the input face.

Some of the various applications of face recognition include driving licenses, immigration, national ID, passport, voter registration, security application, medical records, personal device logon, desktop logon, human-robot-interaction, human-computer-interaction, smart cards etc. Over the last 3 decades, many methods of face recognition have been proposed. Face recognition is such a challenging yet interesting problem that it has attracted researchers who have different backgrounds: pattern recognition, neural networks, computer vision, and computer graphics, hence the literature is vast and diverse. Often, a single system involves techniques motivated by different principles. The usage of a mixture of techniques makes it difficult to classify these systems based on what types of techniques they use for feature representation or classification. To have clear categorization, we follow the holistic and local features approach [2]. Specifically, we have following categories:

1) Holistic matching methods: These methods use the whole face region as a raw input to the recognition system. One of the most widely used representations of the face region is Eigenpictures, which is inherently based on principal component analysis.

2) Feature-based matching methods: Generally, in these methods, local features such as the eyes, nose and mouth are first extracted and their locations and local statistics are fed as inputs into a classifier.

3) Hybrid methods: It uses both local features and whole face region to recognize a face. This method could potentially offer the better of the two types of methods.

Our aim is to extract local features from a given frontal face. The local features are left eye, right eye, nose and mouth. These local features will be extracted manually. Discrete Cosine Transform (DCT) will be applied to each of these local features individually and also to the global features. Finally, the results obtained in both cases will be compared.

Four popular face recognition methods, namely Principle Component Analysis (PCA), Spectroface, Independent Component Analysis (ICA), and Gabor jet are selected and three popular face databases, namely Yale database, Olivetti Research Laboratory (ORL) database and FERRET database, are selected for evaluation in the paper [3]. It proposes to make use of both local features and global features for face recognition and performs experiments in combining two global feature face recognition algorithms, namely, PCA, Spectroface, and two local feature algorithms, namely, Gabor wavelet and ICAs. The experimental results show that the 'rank 1' accuracy ranges from 79.5% to 85.5% and the 'rank 2' accuracy of overall performance for the proposed idea is 92.5%. A face recognition system in which interesting feature points in face are located by Gabor filters is discussed in [4]. Then the filtered image is multiplied with a 2D Gaussian to focus on the center of the face and avoid extracting features at face contour. This Gabor filtered and Gaussian weighted image is then searched for peaks, which are called feature points used for recognition. The accuracy is 83.4% taking all feature points which is better than that obtained in case of single feature strategy.

II. IDEA OF THE PROPOSED SOLUTION

For creating the database standard, BioID database [5] is used. Frontal images are extracted from the database. Frontal face image extraction is carried out in order to reduce the effect of varying backgrounds on the proposed face recognition system. The frontal face image is acquired. From the given frontal image, local features like eyes, nose and mouth region are extracted manually. DCT is applied to each of these features. The DCT coefficients of these regions are stored. These coefficients are then used for comparison. The recognition rates obtained with these local features are compared to the recognition rate obtained when DCT is applied to the global image.

A. Discrete Cosine Transform

DCT is an accurate and robust face recognition system and using certain normalization techniques, its robustness to variations in facial geometry and illumination can be increased [6],[7]. An alternative holistic approach to face recognition is discrete cosine transform. Face normalization techniques were also incorporated in the face recognition system discussed. Namely, an affine transformation was used to correct scale, position, and orientation changes in faces. It was seen that tremendous improvements in recognition rates could be achieved with such normalization. Illumination normalization was also investigated extensively. Various approaches to the problem of compensating for illumination variations among faces were designed and tested, and it was concluded that the recognition rate of the specific system was sensitive to many of these approaches. This was because the faces in the databases used for the tests were uniformly illuminated and these databases contained a wide variety of skin tones. That is, certain illumination normalization techniques had a tendency to make all faces have the same overall grey-scale intensity, and they thus resulted in the loss of much of the information about the individuals' skin tones. A complexity comparison between the DCT and the Karhunen-Loeve transform (KLT) is of interest [1]. In the proposed method, training essentially means computing the DCT coefficients of all the database faces. On the other hand, using the KLT, training entails computing the basis vectors of the transformation. This means that the KLT is more computationally expensive with respect to training. However, once the KLT basis vectors have been obtained, it may be argued that computing the KLT coefficients for recognition is trivial. But this is also true of the DCT, with the additional provision that the DCT may take advantage of very efficient computational algorithms.

DCT is a well-known signal analysis tool used in compression due to its compact representation power [8]. It is known that the KLT is the optimal transform in terms of information packing, however, its data dependent nature makes it infeasible to implement in some practical tasks. Moreover, DCT closely approximates the compact representation ability of the KLT, which makes it a very useful tool for signal representation both in terms of information packing and in terms of computational complexity due to its data independent nature. DCT helps separate the image into parts (or spectral sub-bands) of differing importance (with respect to the image's visual quality). DCT is conceptually similar to Discrete Fourier Transform (DFT), in the way that it transforms a signal or an image from the spatial domain to the frequency domain.

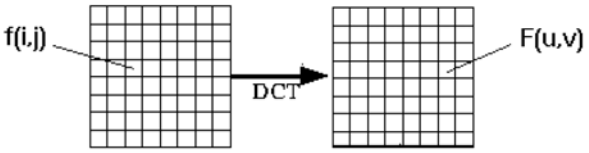


Figure 1. Image transformation from spatial domain to frequency domain

B. Discrete Cosine Transform Encoding

The general equation for a 1D (N data items) DCT is defined as follows:

$$F(u) = \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} A(i) * \cos\left(\frac{u(2i+1)\pi}{2N}\right) * f(i) \qquad (1)$$

where,

$$A(i) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u = 0 \\ 1 & \text{otherwise} \end{cases}$$

f(i) is the input sequence.

The general equation for a 2D (N × M image) DCT is defined as follows:

$$F(u,v) = \sqrt{\frac{2}{N}}\sqrt{\frac{2}{M}}\sum_{i=0}^{N-1} A(i) * \cos\left(\frac{u(2i+1)\pi}{2N}\right) * \sum_{j=0}^{M-1} A(j) * \cos\left(\frac{v(2j+1)\pi}{2M}\right) * f(i,j) \quad (2)$$

where,

$$A(i) = \begin{cases} \frac{1}{\sqrt{2}} & \textit{for } u = 0 \\ 1 & \textit{otherwise} \end{cases}$$

$$A(j) = \begin{cases} \frac{1}{\sqrt{2}} & \textit{for } v = 0 \\ 1 & \textit{otherwise} \end{cases}$$

f(i, j) is the 2D input sequence.

The basic encoding operation of the DCT is as follows:

- The size of the input image is N × M.
- f(i, j) is the intensity of the pixel at x(i, j).
- F(u, v) is the DCT coefficient for the pixel at x(i, j).
- For most images, much of the signal energy lies at low frequencies. These appear in the upper left corner of the DCT.
- Compression is achieved since the lower right values represent higher frequencies, and are often small enough to be neglected with little visible distortion. The output array of DCT coefficients contains integers; these can range from -1024 to 1023.

Computationally, it is easier to implement and also efficient to consider the DCT as a set of basis functions which given a known input array size, for e.g., 8 × 8, can be pre-computed and stored. This involves simply computing values for a convolution mask (8 × 8 window) that get applied. The DCT coefficients are calculated using (2).

III. IMPLEMENTATION STEPS

The database consists of a set of images of 25 people. There are four test images and a single training or registered image. There are 4 different test-methods: GLOBAL, LOCAL, GLOBAL+LOCAL and GLOBAL AND LOCAL. The images are converted to average intensity with respect to the registered image stored in the database, i.e., the images are normalized.

A. Normalization

Since the facial images are captured at different instants of the day or on different days, the intensity for each image may exhibit variations. To avoid these light intensity variations, the test images are normalized so as to have an average intensity value with respect to the registered image. The average intensity value of the registered images is calculated as summation of all pixel values divided by the total number of pixels. Similarly, average intensity value of the test image is calculated. The normalization value is calculated as:

$$\text{Normalization Value} = \frac{\text{Average value of registered image}}{\text{Average value of test image}} \quad (3)$$

This value is multiplied with each pixel of the test image. Thus we get a normalized image having an average intensity with respect to that of the registered image. The entire image is of size 128 × 128 pixels. Upon applying DCT and performing zigzag scanning, we obtain 16384 coefficients, of which 64 coefficients are taken into account while matching.

B. Zigzag Scanning

The purpose of Zigzag Scanning is to:

- Map 128 × 128 to a 1 × 16384 vector.
- Group low frequency coefficients present at the top of the vector.

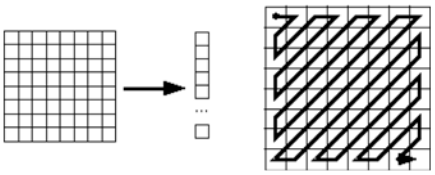


Figure 2. Zigzag scanning

The eye regions are cropped with a size of 16 × 16 pixels. Next, the nose and mouth region are cropped. The mouth region contains the outline of the lips. This region is rectangular in shape. The nose region is 25 × 40 pixels and the mouth region is 30 × 50 pixels. Local features such as eyes, nose and mouth are extracted manually from the given face image. The image is used as an input. The centre eye pixels are located and a region of 16 × 16 pixels is extracted which covers the eye region. The nose region and the mouth region are extracted as shown in Fig. 4. The maximum margin for the nose region is 40 × 25 pixels while that for the mouth region is 50 × 30 pixels. Only the lower portion of nose region, i.e., the region around the nostrils, should be extracted. Similarly, for the mouth region, only the outline of the lips should be considered. Local feature extraction, especially eye extraction, helps reduce the effect of varying characteristics such as pose, expressions etc. on the face recognition system.

Figure 3. Test image

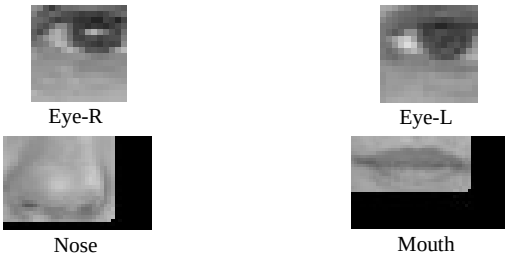


Figure 4. Local feature extraction

After extracting the features, the images are compared using all the methods mentioned above. The comparison is done by taking the Euclidean distance between the test and registered image. For e.g., we take 50 coefficients of test image and 50 coefficients of registered image. The Euclidean distance of each of the 50 coefficients of the test image and 50 coefficients of the registered image are calculated. Add all the Euclidean distance of the 50 coefficients. Let the value of this addition be 'X'. Since there are 15 registered images, each of the test images will be compared with the 15 registered images. Thus we have X_1, X_2, X_3, . . . , X_{15} values. All these 15 values of 'X' are sorted in ascending order, and the one with the minimum value of 'X' is given as the 'rank 1'. The next in the order is given 'rank 2', 'rank 3', till 'rank 15'. This 'rank 1' image is regarded as the best match. If the 'rank 1' image is the same as the input image, the person has been recognized correctly. Thus, the recognition rate is calculated as the ratio of number of images correctly recognized to the total number of images tested. The number of coefficients is varied and the recognition rate is calculated for each of them using the following equation:

$$\text{Recognition rate} = \frac{\text{Number of correctly recognized persons}}{\text{Total number of persons tested}} \quad (4)$$

IV. RESULTS

TABLE I. DCT RESULTS

Features	Recognition Rate
Entire Image	88.25%
Eye-R	87.18%
Eye-L	86.1%
Nose	56.2%
Mouth	52.35%

A. Global Features

Recognition rate using global features, i.e., taking the entire image, has been tabulated as follows:

TABLE II. RECOGNITION RATE USING GLOBAL FEATURES

Global Features	Recognition Rate	
	Without normalized image	*With normalized image*
64 coeffecients	88.25%	92.5%

B. Local Features

Recognition rate using only local features, i.e., taking eyes, nose and mouth templates, has been tabulated as follows:

TABLE III. RECOGNITION RATE USING LOCAL FEATURES

Local Features	Recognition Rate	
	Without normalized image	*With normalized image*
DCT	87.18%	90.2%

C. Using AND Logic

The ranks of both the global feature and local features are compared. If both the ranks are '1' only then is the person accepted, else the person's entry is termed as 'invalid'. Thus the false acceptance rate is zero in this case. The corresponding recognition rate has been tabulated as follows:

TABLE IV. RECOGNITION RATE USING AND LOGIC

AND Logic	Recognition Rate	
	Without normalized image	*With normalized image*
DCT	80.52%	82.35%

D. Combining Local and Global Features

The local templates and global templates are combined together, with different weights assigned to each template. The corresponding recognition rate has been tabulated as follows:

TABLE V. RECOGNITION RATE ON COMBINING LOCAL AND GLOBAL FEATURES

Combination	Recognition Rate	
	Without normalized image	*With normalized image*
DCT	90.4%	94.5%

V. CONCLUSION

When using local features for recognition, the false acceptance rate should be minimized and false rejection rate should be maximized as compared to that of global features. The recognition rate for local features and the recognition rate for global features using DCT is calculated. Comparison between DCT global features and DCT local features is done. The recognition rate improves when images are normalized. When local and global features are combined, DCT gives a relatively high recognition rate.

REFERENCES

[1] Ziad M. Hafed, "Face Recognition Using DCT", International Journal of Computer Vision, 2001, pp. 167-188.

[2] W. Zhao, R. Chellappa, "Face Recognition: A Literature Survey", ACM Computing Surveys, Vol. 35, No. 4, December 2003, pp. 399-458, pp. 9-11.

[3] J. Huang, P. Yuen, J. Lai, "Face Recognition Using Local and Global Features", EURASIP Journal on Applied Signal Processing 2004:4, pp. 530-541.

[4] E. Hjelmas, "Feature-Based Face Recognition", Department of Informatics, University of Oslo, pp. 1-5.

[5] BioID Database, www.bioid.com, October 2007.

[6] M. Tistarelli, L. Akarun, "Report on face state of art", BioSecure, Biometrics for secure Authentication, April 2005, pp. 1-76.

[7] M. Ahmad, T. Natarajan and K. R. Rao, "Discrete Cosine Transform", IEEE Trans. Computers, 1974, pp. 90-94.

[8] H. Ekenel, R. Stiefelhagen, "Block Selection in the Local Appearance-based Face Recognition Scheme" , Interactive Systems Labs, Computer Science Department, University Karlsruhe, Germany, pp. 1-6.

Audio Watermarking with Error Correction

Aman Chadha, Sandeep Gangundi, Rishabh Goel, Hiren Dave, M. Mani Roja
Department of Electronics & Telecommunication
Thadomal Shahani Engineering College
Mumbai, INDIA

***Abstract*— In recent times, communication through the internet has tremendously facilitated the distribution of multimedia data. Although this is indubitably a boon, one of its repercussions is that it has also given impetus to the notorious issue of online music piracy. Unethical attempts can also be made to deliberately alter such copyrighted data and thus, misuse it. Copyright violation by means of unauthorized distribution, as well as unauthorized tampering of copyrighted audio data is an important technological and research issue. Audio watermarking has been proposed as a solution to tackle this issue. The main purpose of audio watermarking is to protect against possible threats to the audio data and in case of copyright violation or unauthorized tampering, authenticity of such data can be disputed by virtue of audio watermarking.**

Keywords- watermarking; audio watermarking; data hiding; data confidentiality.

I. INTRODUCTION

Over the years, there has been tremendous growth in computer networks and more specifically, the internet. This phenomenon, coupled with the exponential increase of computer performance, has facilitated the distribution of multimedia data such as images, audio, video etc. Data transmission has been made very simple, fast and accurate using the internet. However, one of the main problems associated with transmission of data over the internet is that it may pose a security threat, i.e., personal or confidential data can be stolen or hacked in many ways. Publishers and artists, hence, may be reluctant to distribute data over the Internet due to lack of security; copyrighted material can be easily duplicated and distributed without the owner‟s consent. Therefore, it becomes very important to take data security into consideration, as it is one of the essential factors that need attention during the process of data distribution. Watermarks have been proposed as a way to tackle this tough issue. This digital signature could discourage copyright violation, and may help determine the authenticity and ownership of an image.

Watermarking is “the practice of imperceptibly altering a Work to embed a message about that Work” [1]. Watermarking can be used to secretly transmit confidential messages, for e.g., military maps, without the fact of such transmission being discovered. Watermarking, being ideally imperceptible, can be essentially used to mask the very existence of the secret message [6]. In this manner, watermarking is used to create a covert channel to transmit confidential information [5]. Watermarking is an effective means of hiding data, thereby protecting the data from unauthorized or unwanted viewing. Watermarking is becoming increasingly popular, especially for insertion of undetectable identifying marks, such as author or copyright information to the host signal. Watermarking may probably be best used in conjunction with another data-hiding method such as steganography, cryptography etc. Such data-hiding schemes, when coupled with watermarking, can be a part of an extensive layered security approach.

To combat online music piracy, a digital watermark could be added to all recordings prior to release, signifying not only the author of the work, but also the user who has purchased a legitimate copy. Audio watermarking is defined as “the imperceptible, robust and secure communication of data related to the host audio signal, which includes embedding into, and extraction from, the host audio signal” [4]. Digital audio watermarking involves the concealment of data within a discrete audio file. Intellectual property protection is currently the main driving force behind research in this area. Several other applications of audio watermarking such as copyright protection, owner identification, tampering detection, fingerprinting, copy and access control, annotation, and secret communication, are in practice. Other related uses for watermarking include embedding auxiliary information which is related to a particular song, like lyrics, album information, or a hyperlink etc. Watermarking could be used in voice conferencing systems to indicate to others which party is currently speaking. A video application of this technology would consist of embedding subtitles or closed captioning information as a watermark [7].

II. IDEA OF THE PROPOSED SOLUTION

A. *Process of Watermarking*

The block diagram for watermarking is as shown below:

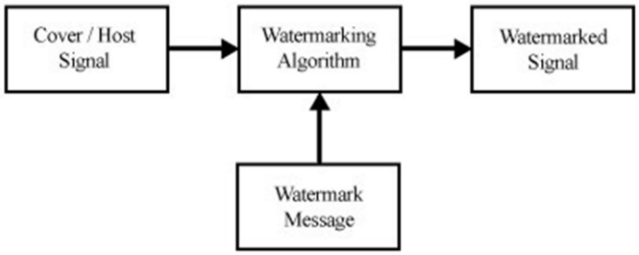


Figure 1. General watermarking block diagram

We can summarize the entire process of hiding and retrieving data as follows:

- Read the data to be hidden.
- Read the cover, i.e., host, in which data is to be hidden.
- Apply watermarking methods on the host.
- Hide the data in the host.
- Retrieve the original data at the receiver end.

B. Mean Square Error

Mean Square Error (MSE) [2], first introduced by C. F. Gauss, serves as an important parameter in gauging the performance of the watermarking system. The following factors justify the choice of MSE as a convenient and extensive standard for performance assessment of various techniques of audio watermarking:

1) Simplicity: It is parameter-free and inexpensive to compute, with a complexity of only one multiply and two additions per sample. It is also memory less, i.e., MSE can be evaluated at each sample, independent of other samples.

2) Clear physical meaning: It is the natural way to define the energy of the error signal. Such an energy measure is preserved even after any orthogonal or unitary linear transformation. The energy preserving property guarantees that the energy of a signal distortion in the transform domain is the same as that in the signal domain.

3) Excellent metric in the context of optimization: The MSE possesses the properties of convexity, symmetry, and differentiability.

4) Used as a convention: It has been extensively employed for optimizing and assessing a wide variety of signal processing applications, including filter design, signal compression, restoration, reconstruction, and classification.

MSE is essentially a signal fidelity measure [14],[15]. The goal of a signal fidelity measure is to compare two signals by providing a quantitative score that describes the degree of similarity/fidelity or, conversely, the level of error/distortion between them. Usually, it is assumed that one of the signals is a pristine original, while the other is distorted or contaminated by errors.

Suppose that x = $\{x_i \mid i = 1, 2, \ldots, N\}$ and y = $\{y_i \mid i = 1, 2, \ldots, N\}$ are two finite-length, discrete signals, for e.g., visual images or audio signals. The MSE between the signals is given by the following formula:

$$MSE(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2 \qquad (1)$$

Where,

N is the number of signal samples.
x_i is the value of the i^{th} sample in x.
y_i is the value of the i^{th} sample in y.

III. Implementation Steps

Audio Watermarking can be implemented in 3 ways:

- Audio in Audio
- Audio in Image
- Image in Audio

A. Audio in Audio

In this method, both the cover file and the watermark file are audio signals. The watermark signal must have fewer samples as compared to those of the cover audio signal. Further, this method can be implemented with the help of two techniques, namely, Interleaving and DCT.

1) Using Interleaving: It is a way to arrange data in a non-contiguous way so as to increase performance [3]. The following example illustrates the process of interleaving:

Original signal:	AAAABBBBCCCCDDDDEEEE
Interleaved signal:	ABCDEABCDEABCDEABCDE

In this technique, the samples of watermark audio are inserted in between the samples of the cover audio file [9],[10]. In terms of complexity, this is the simplest method of audio watermarking.

2) Using Discrete Cosine Transform: This technique is based on the Discrete Cosine Transform (DCT) [11]-[13]. In this technique, we take the DCT of both the cover audio and the watermark audio signals. Upon zigzag scanning, the high frequency DCT coefficients of the cover audio file are replaced with the low frequency DCT coefficients of the watermark audio file. During transmission, the Inverse Discrete Cosine Transform (IDCT) of the final watermarked DCT is taken. In this particular technique, since both the host, i.e., cover and watermark signal are in the audio format, we implement this method using a 1D DCT which is defined by the following equation:

$$F(u) = \sqrt{\frac{2}{N}}\sum_{i=0}^{N-1} A(i) * \cos\left(\frac{u(2i+1)\pi}{2N}\right) * f(i) \qquad (2)$$

Where,

$$A(i) = \begin{cases} \frac{1}{\sqrt{2}} & for\ u = 0 \\ 1 & otherwise \end{cases}$$

f(i) is the input sequence.

B. Audio in Image

This watermarking implementation uses DCT for embedding audio file in an image. Here we take DCT of both the cover image and the watermark audio files. The low frequency coefficients of both the DCT"s are taken. The high frequency coefficients of the DCT of the image are replaced with the low frequency coefficients of the DCT of the

watermark audio file. During transmission, the IDCT of the final watermarked DCT is taken. This technique involves both an audio signal (watermark) and an image (host). Hence, we implement this method using a 1D DCT for the audio signal and a 2D DCT for the image. Equation (2) defines a 1D DCT while the corresponding equation for a 2D DCT is defined by the following equation:

$$F(u,v) = \sqrt{\frac{2}{N}}\sqrt{\frac{2}{M}}\sum_{i=0}^{N-1} A(i) * \cos\left(\frac{u(2i+1)\pi}{2N}\right) * \sum_{j=0}^{M-1} A(j) * \cos\left(\frac{v(2j+1)\pi}{2M}\right) * f(i,j) \quad (3)$$

Where,

$$A(i) = \begin{cases} \frac{1}{\sqrt{2}} & for\ u = 0 \\ 1 & otherwise \end{cases}$$

$$A(j) = \begin{cases} \frac{1}{\sqrt{2}} & for\ v = 0 \\ 1 & otherwise \end{cases}$$

f(i, j) is the 2D input sequence.

C. Image in Audio

In this implementation, as deployed previously, DCT is used for embedding an image in an audio file. Here we take the DCT of both the cover audio and the watermark image files. This is followed by zigzag scanning so as to ascertain the low frequency and high frequency DCT coefficients. The high frequency DCT coefficients of the audio signal are replaced with the low frequency DCT coefficients of the watermark image file. While transmitting, the IDCT of the final watermarked DCT is taken.

Since this method is similar to the „Audio in Image" technique with respect to the parameters involved, i.e., this technique also involves both an audio signal (watermark) and an image (host), hence, we may implement this method using a 1D DCT for the audio signal and a 2D DCT for the image. Equation (2) defines a 1D DCT while, (3) defines a 2D DCT.

IV. RESULTS

A. Audio in Audio

The spectra of the input, i.e., original cover and watermark audio signal are as shown below:

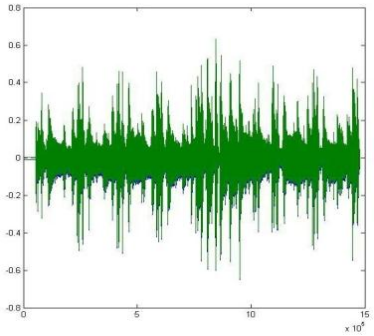


Figure 2. Original cover audio signal

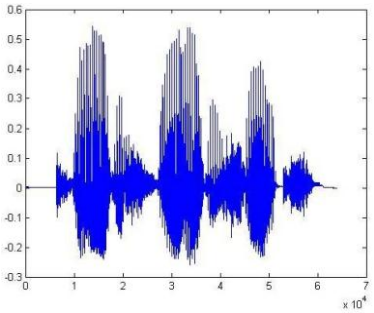


Figure 3. Original watermark audio signal

1) Using Interleaving: The spectra of the output, i.e., watermarked audio signal and the recovered audio signal, obtained by interleaving are as shown below:

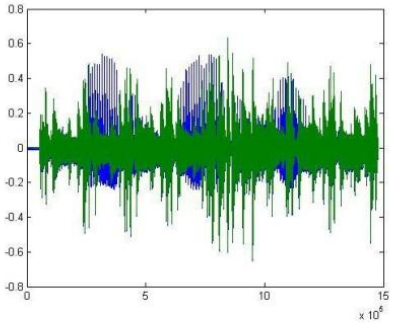


Figure 4. Watermarked audio signal

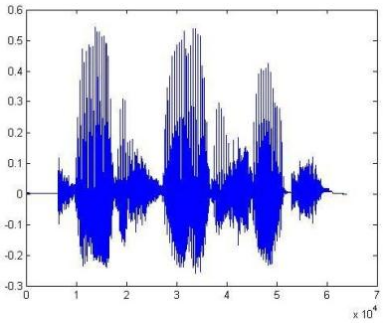


Figure 5. Recovered audio watermark signal

2) Using DCT: The spectra of the output, i.e., watermarked audio signal and the recovered audio signal, obtained by DCT based „Audio in Audio" watermarking are as shown below:

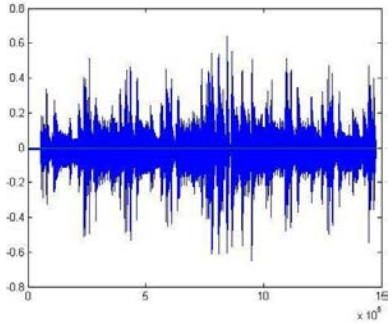


Figure 6. Watermarked audio signal

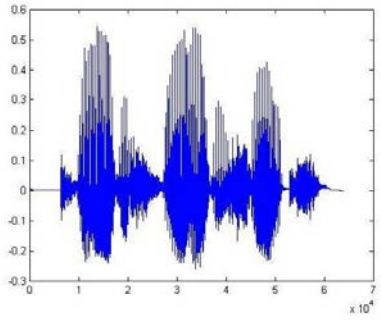


Figure 7. Recovered audio watermark signal

B. Audio in Image

The input, i.e., original cover image and spectrum of the watermark audio signal is as shown below:

Figure 8. Original cover image

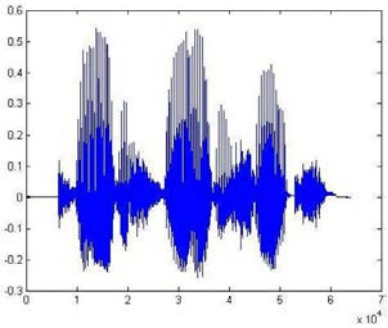


Figure 9. Original watermark audio signal

The output, i.e., watermarked image and the spectrum of the recovered audio signal, obtained by „Audio in Image" watermarking is as shown below:

Figure 10. Watermarked image

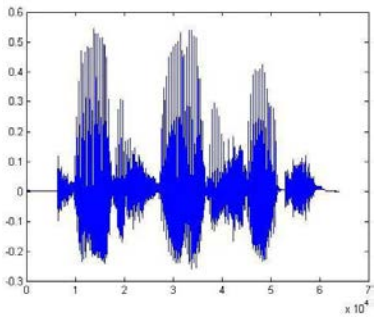


Figure 11. Recovered audio watermark signal

C. Image in Audio

The input, i.e., spectrum of the original cover signal and the watermark image is as shown below:

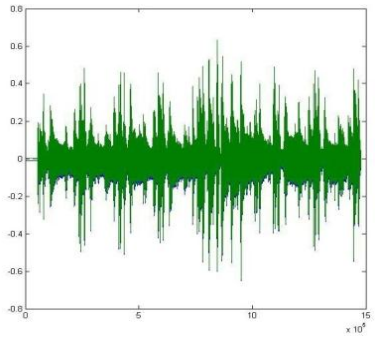


Figure 12. Original cover audio signal

Figure 13. Original watermark image

The output, i.e., spectrum of the watermarked signal and the recovered watermark image, obtained by „Image in Audio" watermarking is as shown below:

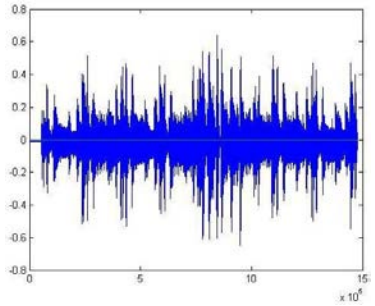


Figure 14. Watermarked image

Figure 15. Recovered watermark

For all practical purposes, it is preferable to have a quantitative measurement to provide an objective judgment of the extracting fidelity. This is done by calculating the MSE in each case. Results, thus obtained, have been tabulated.

TABLE I. CALCULATION OF MSE WITHOUT NOISE

Types	MSE	
	Watermarked signal	*Watermark*
Audio in Audio (Interleaving)	3.5×10^{-4}	0
Audio in Audio (DCT)	3.4×10^{-3}	2.48×10^{-9}
Image in Audio	3.2×10^{-3}	0
Audio in Image	23.3	2.47×10^{-9}

V. ERROR CORRECTION USING HAMMING CODES

During transmission, when data travels through a wireless medium over a long distance, it is highly probable that it may get corrupted due to fluctuations in channel characteristics or other external parameters. Hence, hamming codes, which are Forward Error Correction (FEC) codes, may be used for the purpose of error correction in audio watermarking.

Hamming codes can detect up to two simultaneous bit errors and correct single bit errors; thus, reliable communication is possible when the "hamming distance" between the transmitted and received bit patterns is less than or equal to one [8]. In contrast, simple parity codes cannot correct errors and can only detect an odd number of errors. In this application, we are using the (15, 11) form of the hamming code. The improvement in the quality of received signal is clearly visible from the following waveforms:

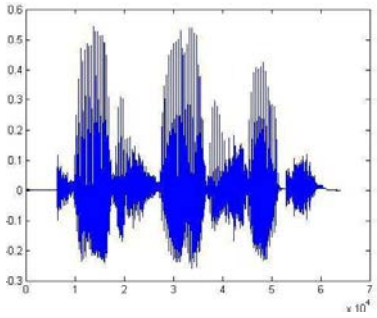


Figure 16. Desired audio signal

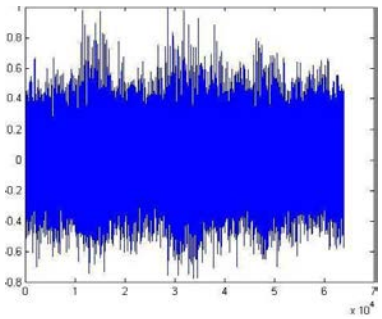


Figure 17. Recovered audio signal without Hamming Code

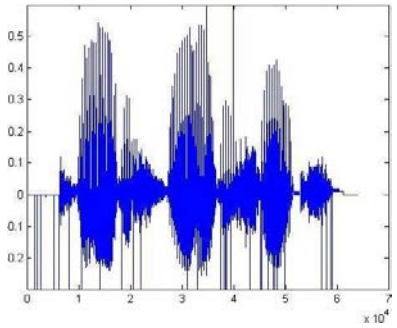


Figure 18. Recovered audio signal with Hamming Code

The corresponding results for Audio Watermarking with noise have been tabulated as follows:

TABLE II. CALCULATION OF MSE WITH NOISE

Types	MSE	
	Watermarked with noise	*Watermark with noise*
Audio in Audio (Interleaving)	0.010	0.010
Audio in Audio (DCT)	0.002	0.008
Image in Audio	0.003	1.92×10^{3}
Audio in Image	323.3	0

VI. CONCLUSION

DCT is an effective and robust algorithm for audio watermarking as the audio signal retrieved is clearly audible. Hamming Codes enable data correction in case of data corruption during transmission and help in recovering the original audio signal.

REFERENCES

[1] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, Digital Watermarking and Steganography, 2nd Edition, Morgan Kaufmann, 2008, p. 31.

[2] Z. Wang and A. C. Bovik, "Mean squared error: love it or leave it? - A new look at signal fidelity measures," IEEE Signal Processing Magazine, Vol. 26, No. 1, pp. 98-117, January 2009.

[3] S. Shiyamala and Dr. V. Rajamani, "A Novel Area Efficient Folded Modified Convolutional Interleaving Architecture for MAP Decoder," International Journal of Computer Applications (IJCA), Vol. 9, No. 9, p. 1, November 2010.

[4] N. Cvejic, Academic Dissertation, Faculty of Technology, University of Oulu, "Algorithms for audio watermarking and steganography," p. 5.

[5] Shahreza S.S. and Shalmani M.T.M., "Adaptive wavelet domain audio steganography with high capacity and low error rate," Proceedings of the IEEE International Conference on Information and Emerging Technologies, (ICIET,,07), pp. 1729-1732, 2007.

[6] Dr. H. B. Kekre and A. A. Archana, "Information hiding using LSB technique with increased capacity," International Journal of Cryptography and Security, Vol. 1, No. 2, p. 1, October 2008.

[7] M. Adya, M.Tech. Thesis, Indian Institute of Technology Kharagpur

(IIT-KGP), "Audio watermark resistant to mp3 compression," p. 13.

[8] D. MacKay, "Information Theory, Inference and Learning Algorithms," Cambridge University Press, September 2003 pp. 218-222.

[9] C. Zeng and Y. Zhong, "An interleaving-decomposition based digital watermarking scheme in wavelet domain," 8th International Conference on Signal Processing, Vol. 4, Nov. 2006.

[10] P. Agarwal and B. Prabhakaran, "Reliable Transmission of Audio Streams in Lossy Channels Using Application Level Data Hiding," Journal of Multimedia, Vol. 3, No. 5, December 2008, pp. 1-8.

[11] J. Liu and Z. Lu, "A Multipurpose Audio Watermarking Algorithm Based on Vector Quantization in DCT Domain," World Academy of Science, Engineering and Technology (WASET), Issue 55, July 2009, pp. 399-404.

[12] R. Ravula, M.S. Thesis, Department of Electrical and Computer Engineering, Louisiana State University and Agricultural and Mechanical College, "Audio Watermarking using Transformation Techniques," pp. 8-51.

[13] Y. Yan, H. Rong, and X. Mintao, "A Novel Audio Watermarking Algorithm for Copyright Protection Based on DCT Domain," Second International Symposium on Electronic Commerce and Security, China, May 2009, pp. 184 - 188.

[14] G. Casella and E.L. Lehmann, Theory of Point Estimation. New York: Springer-Verlag, 1999.

[15] T. N. Pappas, R. J. Safranek and J. Chen, "Perceptual criteria for image quality evaluation," in Handbook of Image and Video Processing, 2nd ed., May 2005.

Proceedings of the 2011 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)
IEEE Xplore: CFB1120J-ART; ISBN: 978-1-61284-694-1; Print Version: CFB1120J-PRT; ISBN: 978-1-61284-766-5

Design and Simulation of an 8-bit Dedicated Processor for calculating the Sine and Cosine of an Angle using the CORDIC Algorithm

Aman Chadha[1,a], Divya Jyoti[2,b] and M. G. Bhatia[3,c]
[1,2] Thadomal Shahani Engineering College, Bandra (W), Mumbai, INDIA
[3] Ameya Centre for Robotics and Embedded Technology, Andheri (W), Mumbai, INDIA
[a] aman.x64@gmail.com, [b] dj.rajdev@gmail.com, [c] mgbhatia@acret.in

Abstract - **This paper describes the design and simulation of an 8-bit dedicated processor for calculating the Sine and Cosine of an Angle using CORDIC Algorithm (COordinate Rotation DIgital Computer), a simple and efficient algorithm to calculate hyperbolic and trigonometric functions. We have proposed a dedicated processor system, modeled by writing appropriate programs in VHDL, for calculating the Sine and Cosine of an angle. System simulation was carried out using ModelSim 6.3f and Xilinx ISE Design Suite 12.3. A maximum frequency of 81.353 MHz was reached with a minimum period of 12.292 ns. 126 (3%) slices were used. This paper attempts to survey the existing CORDIC algorithm with an eye towards implementation in Field Programmable Gate Arrays (FPGAs). A brief description of the theory behind the algorithm and the derivation of the Sine and Cosine of an angle using the CORDIC algorithm has been presented. The system can be implemented using Spartan3 XC3S400 with Xilinx ISE 12.3 and VHDL.**

Keywords - **CORDIC, VHDL, dedicated processor, datapath, finite state machine.**

I. INTRODUCTION

Over the years, the field of Digital Signal Processing (DSP) has been essentially dominated by Microprocessors. This is mainly because of the fact that they provide designers with the advantages of single cycle multiply-accumulate instruction as well as special addressing modes [4]. Although these processors are cheap and flexible, they are relatively less time-efficient when it comes to performing certain resource-intensive signal processing tasks, e.g., Image Compression, Digital Communication and Video Processing. However as a direct consequence of rapid advancements in the field of VLSI and IC design, special purpose processors with custom-architectures are designed to perform certain specific tasks. They need fewer resources and are less complex than their general purpose counterparts. Instructions for performing a task are hardwired into the processor itself, i.e., the program is built right into the microprocessor circuit itself [2]. Due to this, the execution time of the program is considerably less than that if the instructions are stored in memory. Emerging high level hardware description and synthesis technologies in conjunction with Field Programmable Gate Arrays (FPGAs) have significantly lowered the threshold for hardware development as opportunities exist to integrate these technologies into a tool for exploring and evaluating micro-architectural designs [4]. Because of their advantage of real-time in-circuit reconfigurability, FPGAs based processors are flexible, programmable and reliable [1]. Thus, higher speeds can be achieved by these customized hardware solutions at competitive costs. Also, various simple and hardware-efficient algorithms exist which map well onto these chips and can be used to enhance speed and flexibility while performing the desired signal processing tasks [1],[2],[3].

One such simple and hardware-efficient algorithm is COordinate Rotation DIgital Computer (CORDIC) [5]. Primarily developed for real-time airborne computations, it uses a unique computing technique highly suitable for solving the trigonometric relationships involved in plane co-ordinate rotation and conversion from rectangular to polar form. John Walther extended the basic CORDIC theory to provide solution to and implement a diverse range of functions [7]. It comprises a special serial arithmetic unit having three shift registers, three adders/subtractors, Look-Up Table (LUT) and special interconnections. Using a prescribed sequence of conditional additions or subtractions, the CORDIC arithmetic unit can be designed to solve either of the following equations:

$$\begin{aligned} Y' &= K\left(Y\cos\lambda + X\sin\lambda\right) \\ X' &= K\left(X\cos\lambda - Y\sin\lambda\right) \end{aligned} \quad (1)$$

Where, K is a constant.

By making slight adjustments to the initial conditions and the LUT values, it can be used to efficiently implement trigonometric, hyperbolic, exponential functions, coordinate transformations etc. using the same hardware. Since it uses only shift-add arithmetic, the VLSI implementation of such an algorithm is easily achievable [4].

II. CORDIC ALGORITHM

The CORDIC algorithm is an iterative technique based on the rotation of a vector which allows many transcendental and trigonometric functions to be calculated. The key aspect of this method is that it is achieved using only shifts, additions/subtractions and table look-ups which map well into hardware and are ideal for FPGA implementation. The CORDIC algorithms presented in this paper are well known in the research and super-computing circles.

A. Algorithm Fundamentals

Vector rotation is the first step to obtain the trigonometric functions. It can also be used for polar to rectangular and vice-versa conversions, for vector magnitude, and as a building block in certain transforms such as the Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT). The algorithm is derived from Givens [6] rotation as follows:

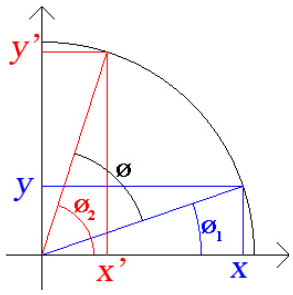


Fig. 1. Illustration of the CORDIC algorithm

In Fig. 1, the diagonal blue line is at an angle ϕ_1 above the horizontal. The diagonal red line is actually the blue line rotated anti-clockwise by an angle ϕ. The new X and Y values are related to the old X and Y values as follows:

$$\begin{aligned} x' &= x\cos\phi - y\sin\phi \\ y' &= y\cos\phi + x\sin\phi \end{aligned} \tag{2}$$

For CORDIC, the final angle ϕ_2 the angle whose sine or cosine we want to calculate and initial angle ϕ_1 is set to a convenient value such as 0. Rather than rotating from ϕ_1 to ϕ_2 in one full sweep, we move in steps with careful choice of step values. Rearranging (2) gives us:

$$\begin{aligned} x' &= \cos\phi\cdot\left[x - y\tan\phi\right] \\ y' &= \cos\phi\cdot\left[y + x\tan\phi\right] \end{aligned} \tag{3}$$

Restricting the rotation angles such that $\tan\phi = \pm 2^{-i}$, transforms the multiplication by the tangent term to a simple shift operation [1]. Arbitrary angles of rotation are obtained by successively performing smaller elementary rotations. If i, the decision at each iteration, is which direction to rotate rather than whether to rotate or not, then $\cos(\delta_i)$ is constant as $\cos(\delta_i) = \cos(-\delta_i)$. Then the iterative rotation can be expressed as:

$$\begin{aligned} x_{i+1} &= K_i\left[x_i - y_i\cdot d_i\cdot 2^{-i}\right] \\ y_{i+1} &= K_i\left[y_i + x_i\cdot d_i\cdot 2^{-i}\right] \end{aligned} \tag{4}$$

$$\text{Where, } K_i = \cos(\tan^{-1}2^{-i}) = \frac{1}{\sqrt{1+2^{-i}}} = \left(\sqrt{1+2^{-i}}\right)^{-1}$$

$$d_i = \pm 1$$

Removing the scale constant from the iterative equations yields a shift-add algorithm for vector rotation. The product of the K_i's can be applied elsewhere in the system or treated as part of a system processing gain. That product approaches 0.6073 as the number of iterations reaches infinity. Therefore, the rotation algorithm has a gain, $A_n \approx 1.65$. The exact gain depends on the number of iterations, and follows the following equation:

$$A_n = \prod_n \sqrt{1 + 2^{-2i}} \tag{5}$$

The angle of a composite rotation is realized by the sequence of the directions of the elementary rotations. That sequence can be represented by a decision vector. The set of all possible decision vectors is an angular measurement system based on binary arctangents. Conversions between this angular system and any other can easily be accomplished using a LUT. A better conversion method uses an additional adder-subtractor that accumulates the elementary rotation angles post iteration. The angle accumulator adds a third difference equation to the CORDIC algorithm:

$$z_{i+1} = z_i - d_i\cdot\tan^{-1}\left(2^{-i}\right) \tag{6}$$

As discussed above, when the angle is in the arctangent base, this extra element is not needed. The CORDIC rotator is normally operated in one of two modes, i.e., the Rotation mode and the Vectoring mode.

B. Rotation Mode

The first mode of operation, called rotation by Volder [5],[4], rotates the input vector by a specified angle (given as an argument). Here, the angle accumulator is initialized with the desired rotation angle. The rotation decision based on the sign of the residual angle is made to diminish the magnitude of the residual angle in the angle accumulator. If the input angle is already expressed in the binary arctangent base, the angle accumulator is not needed [4],[1]. The equations for this are:

$$\begin{aligned} x_{i+1} &= x_i - y_i\cdot d_i\cdot 2^{-i} \\ y_{i+1} &= y_i + x_i\cdot d_i\cdot 2^{-i} \\ z_{i+1} &= z_i - d_i\cdot\tan^{-1}\left(2^{-i}\right) \end{aligned} \tag{7}$$

$$\text{Where, } d_i = \begin{cases} -1 & \text{if } z_i < 0 \\ +1 & \text{otherwise} \end{cases}$$

$$\begin{aligned} x_n &= A_n[x_0\cos z_0 - y_0\sin z_0] \\ y_n &= A_n[y_0\cos z_0 + x_0\sin z_0] \\ z_n &= 0 \\ A_n &= \prod_n \sqrt{1 + 2^{-2i}} \end{aligned} \tag{8}$$

C. Vectoring Mode

In the vectoring mode, the CORDIC rotator rotates the input vector through whatever angle is necessary to align the result vector with the x axis. The result of the

vectoring operation is a rotation angle and the scaled magnitude i.e. the x component of the original vector. The vectoring function works by seeking to minimize the y component of the residual vector at each rotation. The sign of the residual y component is used to determine which direction to rotate next. When initialized with zero, accumulator contains the traversed angle at the end of the iterations [4]. The equations in this mode are:

$$\begin{aligned} x_{i+1} &= x_i - y_i \cdot d_i \cdot 2^{-i} \\ y_{i+1} &= y_i + x_i \cdot d_i \cdot 2^{-i} \\ z_{i+1} &= z_i - d_i \cdot \tan^{-1}\left(2^{-i}\right) \end{aligned} \quad (9)$$

Where, $d_i = \begin{cases} +1 & \text{if } y_i < 0 \\ -1 & \text{otherwise} \end{cases}$

Then:

$$\begin{aligned} x_n &= A_n \sqrt{x_0^2 + y_0^2} \\ y_n &= 0 \\ z_n &= z_0 + \tan^{-1}\left(\frac{y_0}{x_0}\right) \\ A_n &= \prod_n \sqrt{1 + 2^{-2i}} \end{aligned} \quad (10)$$

The CORDIC rotation and vectoring algorithms as stated are limited to rotation angles between $-\pi/2$ and $\pi/2$. For composite rotation angles larger than $\pi/2$, an additional rotation is required [1]. Volder [4] describes an initial rotation of $\pm \pi/2$. This gives the correction iteration:

$$\begin{aligned} x' &= -d \cdot y \\ y' &= d \cdot x \\ z' &= z + d \cdot \frac{\pi}{2} \end{aligned} \quad (11)$$

Where, $d_i = \begin{cases} +1 & \text{if } y < 0 \\ -1 & \text{otherwise} \end{cases}$

There is no growth for this initial rotation. Alternatively, an initial rotation of either π or 0 can be made, avoiding the reassignment of the x and y components to the rotator elements. Again, there is no growth due to the initial rotation:

$$\begin{aligned} x' &= d \cdot x \\ y' &= d \cdot y \\ z' &= \begin{cases} z & \text{if } d = 1 \\ z - \pi & \text{if } d = -1 \end{cases} \end{aligned} \quad (12)$$

Where, $d_i = \begin{cases} -1 & \text{if } x < 0 \\ +1 & \text{otherwise} \end{cases}$

Both reduction forms assume a modulo 2π representation of the input angle. The second reduction may be more convenient when wiring is restricted, as is often the case with FPGAs.

D. Evaluation of Sine and Cosine using CORDIC

In rotational mode the sine and cosine of the input angle can be computed simultaneously. Setting the y component of the input vector to zero reduces the rotation mode result to:

$$\begin{aligned} x_n &= A_n \cdot x_0 \cos z_0 \\ y_n &= A_n \cdot x_0 \sin z_0 \end{aligned} \quad (13)$$

If x_0 is equal to $1/A_n$, the rotation produces the unscaled sine and cosine of the angle argument, z_0. Very often, the sine and cosine values modulate a magnitude value. Using other techniques (e.g., a LUT) requires a pair of multipliers to obtain the required modulation. The algorithm performs the multiply as part of the rotation operation, and therefore eliminates the need for a pair of explicit multipliers. The output of the CORDIC rotator is scaled by the rotator gain. If the gain is not acceptable, a single multiply by the reciprocal of the gain constant placed before the CORDIC rotator will yield unscaled results [1].

E. Advantages

- Number of gates required in hardware implementation on an FPGA, are minimum. Thus, hardware complexity is greatly reduced compared to other processors such as DSP multipliers. Hence, it is relatively simple in design.
- Due to reduced hardware requirement, cost of a CORDIC hardware implementation is less as only shift registers, adders and look-up table (ROM) are required.
- Delay involved during processing is comparable to that of a division or square-rooting operation.
- No multiplication and only addition, subtraction and bit-shifting operation ensures simple VLSI implementation.
- Either if there is an absence of a hardware multiplier (e.g. microcontroller, microprocessor) or there is a necessity to optimize the number of logic gates (e.g. FPGA), CORDIC is the preferred choice [4].

F. Applications

- The algorithm was basically developed to offer digital solutions to the problems of real-time navigation in B-58 bomber [5].
- This algorithm finds use in 8087 Math coprocessor, the HP-35 calculator [8], radar signal processors [8] and robotics.
- CORDIC algorithm has also been described for the calculation of DFT, DHT, Chirp Z-transforms, filtering, Singular value decomposition and solving linear systems [4].

- Most calculators, especially the ones built by Texas Instruments and Hewlett-Packard use CORDIC algorithm for calculation of transcendental functions.

III. SYSTEM ARCHITECTURE

In this paper, the FPGA implementation of simple 8-bit dedicated processor for calculating the sine and cosine of an angle using CORDIC Algorithm is presented. The processor was implemented by using Xilinx ISE Design Suite 12.3 and VHDL. Fig. 2 shows functional block diagram of our 8-bit processor. It mainly consists of an 8-bit multiplexers, registers, arithmetic logic unit (ALU), tri-state buffer, comparator, and control unit. The logic circuit for dedicated microprocessor is divided into two parts: the datapath unit and control unit [9].

Input of two registers can be either from an external data input or from the output of ALU unit. Two control signals ln_X and ln_Y select which of two sources are to be loaded into registers. Two control signals XLoad and YLoad load a value into respective registers. Bottom multiplier determines the source of two operands of ALU. This allows the selection of one of the two subtraction operations X-Y or Y-X. A comparator unit is used to test condition of equal to or greater than and it accordingly generates status signals. Tristate buffer is used for outputting result from register X.

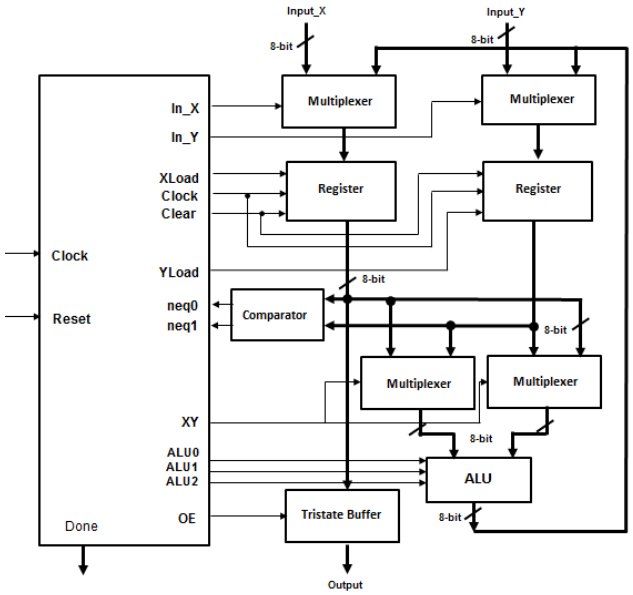


Fig. 2. Functional block diagram of the 8-bit processor

A. Datapath Unit

Datapath is responsible for the actual execution of all data operations performed by the dedicated processor [9]. Fig. 3 shows the datapath unit for the 8-bit dedicated processor.

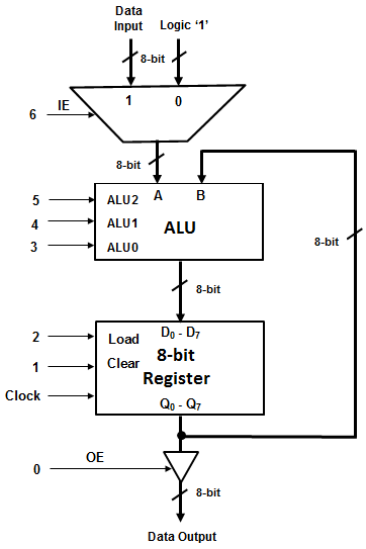


Fig. 3. A simple, general datapath circuit for the dedicated microprocessor

B. Control Unit

Fig. 4 shows the block diagram of control unit and Fig. 5 shows the corresponding state diagram.

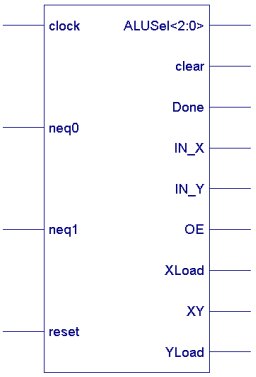


Fig. 3. Block diagram of the control unit

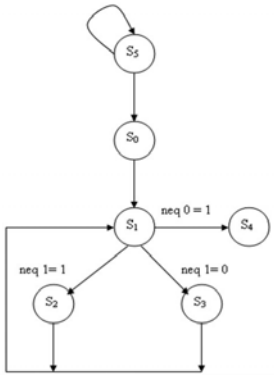


Fig. 4. State diagram of the control unit

Control signals are generated by the control unit which is modelled as a finite state machine with 6 states say S_0-S_5. There are 9 control signals which form control word and control the operation of datapath, as per the following table:

TABLE I
CONTROL SIGNAL STATUS DURING DIFFERENT STATES

ln_X	ln_Y	XLoad	YLoad	XY	Clear
1	1	1	1	0	0
0	0	0	0	0	0
0	0	1	0	1	0
0	0	0	1	0	0
0	0	0	0	0	0
1	1	0	0	0	1

ln_X	ln_Y	ALU(0,1,2)	OE	Done	State
1	1	101	0	0	S_0
0	0	101	0	0	S_1
0	0	101	0	0	S_2
0	0	101	0	0	S_3
0	0	101	1	1	S_4
1	1	101	0	0	S_5

If Reset = '1' then state S_5 occurs. In this state registers are initialized to '0' by asserting the Clear signal. If Reset = '0' then at rising edge of clock, the state is upgraded from S_5 to state S_0. During the state S_0 two inputs are loaded in two registers. After completion of state S_0, state S_1 is reached. In this state output of registers is checked in comparator for equality and greater than conditions. If both values are same, state S_4 occurs else S_2 or S_3 will continue depending on status of signal neq1. State S_1 is repeated again. Default state is S_5.

IV. IMPLEMENTATION AND VERIFICATION

All the units in dedicated processor were designed. These units were described in VHDL-modules and synthesized using ISE Design Suite 12.3. ModelSim simulator was used to verify the functionalities of each unit. Finally all the units were combined together and once again tested by using ModelSim simulator. Fig. 6 shows the RTL schematic of the CORDIC processor generated from Xilinx ISE.

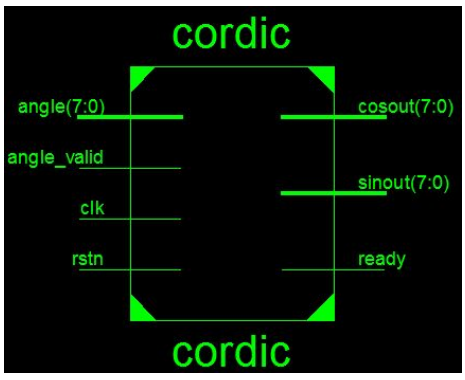


Fig. 6. RTL schematic of the CORDIC processor

A. Datapath Unit

Simulation result of datapath is shown in Fig. 7.

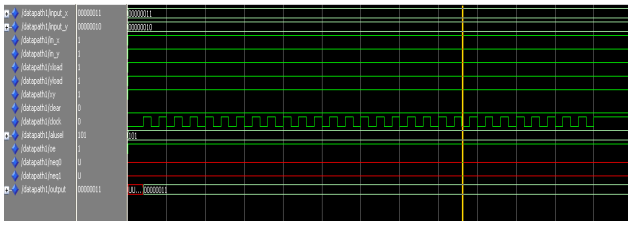


Fig. 7. Datapath unit simulation

The following table shows the synthesis report of the datapath unit:

TABLE II
SYNTHESIS REPORT OF THE DATAPATH UNIT

Number of Slices	61 (31%)
Maximum Frequency	114.05 MHz
Minimum Period	8.76 ns

B. Control Unit

Simulation result of the control unit is shown in Fig. 8.

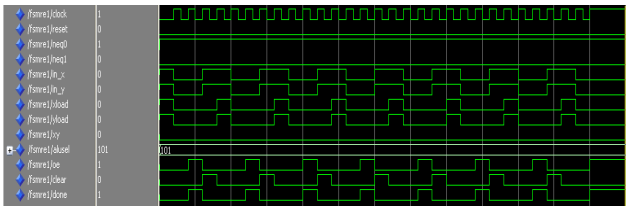


Fig. 8. Control unit simulation

The following table shows the synthesis report of the control unit:

TABLE III
SYNTHESIS REPORT OF THE CONTROL UNIT

Number of Slices	4 (2%)
Maximum Frequency	264.34 MHz
Minimum Period	3.78 ns

C. Dedicated CORDIC Processor

Once the datapath unit and control unit were simulated, they were combined and a dedicated processor was constructed. Simulation shows the CORDIC calculation operation of the Sine and Cosine of an angle. Simulation result for the dedicated processor is as shown in Fig. 9.

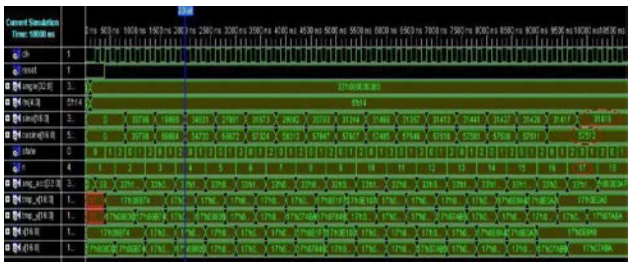


Fig. 9. Test-Bench waveforms indicating the Sine and Cosine output, obtained after the CORDIC Core Simulation

The following table shows the synthesis report of the 8-bit dedicated processor:

TABLE IV
SYNTHESIS REPORT OF THE 8-BIT DEDICATED PROCESSOR

Logic Utilization	Used	Available	Utilization
Number of Slices	126	3584	3%
Number of Slice Flip Flops	58	7168	0%
Number of 4 input LUTs	238	7168	3%
Number of bonded IOBs	28	141	19%
Number of GCLKs	1	8	12%

For a Speed Grade of -5, the minimum period required is 12.292 ns, which corresponds to a maximum frequency of 81.353 MHz. The minimum input arrival time before clock and maximum output required time after clock are 9.657 ns and 8.133 ns respectively.

The following table shows the comparison between the actual Sine and Cosine values and the ones obtained from Test-Bench analysis in VHDL.

TABLE V
COMPARISON OF SUCCESSIVE ANGLE ROTATION VALUES

Angle (A)	Rotation	sin(A) (Actual)	sin(A) (Test-Bench)	Error
0.000000	5	0.00000000	0.01483516	-1.4835e-002
0.000000	10	0.00000000	0.00117259	-1.1725e-003
0.000000	15	0.00000000	0.00001292	-1.2922e-005
0.000000	20	0.00000000	-0.00000043	4.2874e-007
0.523599	5	0.50000000	0.48362630	1.6373e-002
0.523599	10	0.50000000	0.49892865	1.0713e-003
0.523599	15	0.50000000	0.50003905	-3.9047e-005
0.523599	20	0.50000000	0.50000106	-1.0561e-006
1.000000	5	0.84147098	0.80881306	3.2657e-002
1.000000	10	0.84147098	0.84080033	6.7065e-004
1.000000	15	0.84147098	0.84149350	-2.2515e-005
1.000000	20	0.84147098	0.84147186	-8.7478e-007
3.141593	5	0.00000000	-0.01483516	1.4835e-002
3.141593	10	0.00000000	-0.00117259	1.1725e-003
3.141593	15	0.00000000	-0.00001292	1.2922e-005
3.141593	20	0.00000000	0.00000043	-4.2874e-007

Angle (A)	Rotation	cos(A) (Actual)	cos(A) (Test-Bench)	Error
0.000000	5	1.00000000	0.99988995	1.1004e-004
0.000000	10	1.00000000	0.99999931	6.8748e-007
0.000000	15	1.00000000	1.00000000	8.3498e-011
0.000000	20	1.00000000	1.00000000	9.2037e-014
0.523599	5	0.86602540	0.87527459	-9.2491e-003
0.523599	10	0.86602540	0.86664307	-6.1766e-004
0.523599	15	0.86602540	0.86600286	2.2545e-005
0.523599	20	0.86602540	0.86602479	6.0975e-007
1.000000	5	0.54030231	0.58806584	-4.7763e-002
1.000000	10	0.54030231	0.54134537	-1.0430e-003
1.000000	15	0.54030231	0.54026724	3.5067e-005
1.000000	20	0.54030231	0.54030094	1.3623e-006
3.141593	5	-1.0000000	-0.99988995	-1.1004e-004
3.141593	10	-1.0000000	-0.99999931	-6.8748e-007
3.141593	15	-1.0000000	-1.00000000	-8.3498e-011
3.141593	20	-1.00000000	-1.00000000	-9.2037e-014

V. CONCLUSION

We have successfully simulated an 8-bit dedicated processor for calculating the Sine and Cosine of an angle, on ModelSim simulator using the VHDL language. Our processor has six main components namely, control unit, multiplexer unit, ALU unit, register unit, tristate buffer unit and comparator. Our dedicated processor has a maximum frequency of 81.353 MHz was reached with a minimum period of 12.292 ns. 126 (3%) slices were used. Our System can be implemented on Xilinx Spartan 3 XC3S400 using ISE Design Suite 12.3 and VHDL language. Our dedicated processor has a distinct advantage over a general purpose processor, since it repeatedly performs same task its design is more efficient and consumes less resources and is less time intensive.

REFERENCES

[1] R. Andraka, "A survey of CORDIC algorithms for FPGA based computers," *Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays*, pp. 191 – 200.

[2] V. Sharma, *FPGA Implementation of EEAS CORDIC based Sine and Cosine Generator*, M. Tech Thesis, Dept. of Electronics and Communication Engineering, Thapar University, Patiala, 2009.

[3] S. Panda, *Performance Analysis and Design of a Discrete Cosine Transform Processor using CORDIC Algorithm*, M. Tech Thesis, Dept. of Electronics and Communication Engineering, NIT Rourkela, Rourkela, Orissa, 2010.

[4] R. K. Jain, B. Tech Thesis, *Design and FPGA Implementation of CORDIC-based 8-point 1D DCT Processor*, NIT Rourkela, Rourkela, Orissa, 2011.

[5] J. Volder, "The CORDIC Trigonometric Computing Technique," *IRE Transactions on Electronic Computing*, Vol EC-8, Sept 1959, pp. 330-334.

[6] F. Ling, "Givens rotation based least squares lattice and related algorithms," *IEEE Transactions on Signal Processing*, Jul 1991, pp. 1541 – 1551.

[7] J. S. Walther, "A unified algorithm for elementary functions," *Proceedings of the Spring Joint Computer Conference*, 1971, pp. 379-385.

[8] R. Andraka. "Building a High Performance Bit-Serial Processor in an FPGA," *Proceedings of Design SuperCon*, Jan 1996, pp. 1-2.

[9] E. O. Hwang, *Digital Logic and Microprocessor Design with VHDL*, Thomson/Nelson, 2006, pp. 379-413, pp. 290-311.