IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 14, Issue 3 (Sep. - Oct. 2013), PP 79-88
www.iosrjournals.org

Face Recognition System under Varying Lighting Conditions

P.Kalaiselvi[1], S.Nithya[2]
[1](*Asst. Professor, Department of ECE, NSN College of Engineering and Technology, Karur, Tamilnadu, India)*
[2] *(Asst. Professor, Department of ECE, NSN College of Engineering and Technology, Karur, Tamilnadu, India)*

Abstract: *Making recognition more reliable under uncontrolled lighting conditions is one of the most important challenges for practical face recognition systems. Other recognition systems don't nullify most of the lighting variations. We tackle this by combining the strengths of robust illumination normalization, local texture-based face representations, distance transform based matching and kernel based feature extraction and multiple feature fusion. We present a simple and efficient preprocessing chain that eliminates most of the effects of changing illumination while still preserving the essential appearance details that are needed for recognition. We introduce Local Ternary Pattern (LTP), a generalization of the Local Binary Pattern (LBP) local texture descriptor less sensitive to noise. We further increase robustness by introducing Phase Congruency. The resulting method provides a face verification rate of 88.1% at 0.1% false accept rate. Experiments show that our preprocessing method outperforms several existing preprocessors for a range of feature sets, data sets and lighting conditions. We simulate this project using MATLAB software.*

Keywords: *Recognition, Illumination normalization, Local texture based face representations, Local Binary Pattern, Local Ternary Pattern .*

I. INTRODUCTION

Our facial recognition system will save our time & money, as well as hassles involved with other biometric or ID card/Password based technologies. Our face recognition technology uses faces as unique verification information. We offer facial recognition system that works in a wide range of operating environment from individual home environment to most common public places. Face Recognition System focuses on technology of contactless man and machine interface. Our no human touch biometric face detection and face verification system gives high accuracy and reliable security of our need. Our face identification systems are based on easy to use, flexible. Our face authentication methods are useful in

- Access Control
- Time Attendance System
- Visitor Management System

The parameter that motivated us to do this project is that it can able to verify an image obtained even in very poor lighting. Even there are many changes in the lighting of the images, the proposed method is able to identify or recognize the input image.

Face is our primary focus of attention for conveying identity. Detection of faces in a digital image has gained much importance in the last decade, with application in many fields. Automating the process to a computer requires the use of various image processing techniques. Face detection is defined as to isolate human faces from their background and exactly locate their position in an image. Generally, a facial recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source [1]. One of the ways to do this is by comparing selected facial features from the image and a facial database. It is typically used in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems.

Our paper focuses mainly on the issue of robustness to lighting variations. For example, a face verification system for a portable device should be able to verify a client at any time and in any place. Unfortunately, facial appearance depends strongly on the ambient lighting and as emphasized by the recent Face Recognition Vendor Test (FRVT) and Face Recognition Grand Challenge (FRGC) trials. This remains one of the major challenges for current face recognition systems.

So we include a pre-processing chain to eliminate lighting variations. The pre-processing chain includes Gamma correction, DOG filtering, masking and contrast equalization process. This is further followed by Local Binary Patterns and Local Ternary Patterns to make it more clear for verification process. The image clearance is further improved by using distance transform similarity metric.

II. OVERVIEW OF THE PROJECT

A facial recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. One of the ways to do this is by comparing selected facial features from the image and a facial database.

A face recognition system recognizes an individual by matching the input image against images of all users in a database and finding the best match. Face recognition has received significant attention in the last 15 years, due to the increasing number of commercial and law enforcement applications requiring reliable personal authentication and the availability of low-cost recording devices [2]. Depending on the application, a face recognition system can be working either on identification or verification mode.

In a face identification application, the system recognizes an individual by matching the input image against images of all users in a database and finding the best match. In face verification application the user claims an identity and the system accepts or rejects her claim by matching the input image against the image that corresponds to this specific identity, which can be stored either in a database or an identification card (e.g. smart card).

Most of the method were initially developed with face images collected under relatively well-controlled conditions and in practice they have difficulty in dealing with the range of appearance variations that commonly occur in unconstrained natural images due to illumination, facial expression, aging, partial occlusions, etc.

This paper deals with the issue of robustness to lighting variations. Traditional approaches for dealing with this issue broadly classified into three categories such as

2.1 Appearance Based Recognition system

Many approaches to object recognition and to computer graphics are based directly on images without the use of intermediate 3D models [3]. Most of these techniques depend on a representation of images that induces a vector space structure and, in principle, requires dense correspondence. Appearance-based approaches represent an object in terms of several object views.

While template-matching methods rely on a predefined template or model, appearance-based methods use large numbers of examples depicting different variations. Face detection can be viewed as a pattern classification problem with two classes: “face” and “non-face”. The “non-face” class contains images that may depict anything that is not a face, while the “face” class contains all face images. Statistical analysis and machine learning techniques are employed to discover the statistical properties or probability distribution function of the pixel brightness patterns of images belonging in the two classes. To detect a face in an input image, the whole image is scanned and image regions are identified as “face” or “non face” based on these probability functions. Well-known appearance-based methods used for face detection are Eigen faces, LDA, neural networks, support vector machines and hidden Markov models.

2.2 Normalization Based Recognition System

Normalization based approaches seek to reduce the image to a more canonical form in which the illumination variations are suppressed [3]. The normalized image is robust to translation, rotation and scaling. This allows us to recognize subjects based on their normalized images, since the variance of the normalized images of the same subject is much lower than the variance of the original images of that subject.

2.3 Utility Metric and Packet Loss Modeling

This approach extracts illumination-insensitive feature sets directly from the given image. These feature sets range from geometrical features to image derivative features such as edge maps, Local Binary Patterns (LBP), Gabor wavelets, and local autocorrelation filters. Although such features offer a great improvement on raw gray values, their resistance to the complex illumination variations that occur in real-world face images is still quite limited. For example, even though LBP features are completely invariant to monotonic global gray-level transformations, their performance degrades significantly under changes of lighting direction and shadowing.

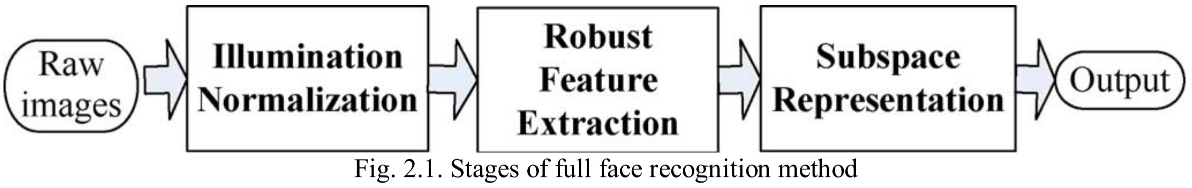


Fig. 2.1. Stages of full face recognition method

III. FACE RECOGNITION SYSTEM

3.1 Illumination Normalization

The source of variation in facial appearance is the illumination of the face. The majority of the techniques proposed to cope with this problem exploit the low dimensionality of the face space under varying illumination conditions. They either use several images of the same person recorded under varying illumination conditions or rely on the availability of 3D face models and different maps to generate novel views. The main shortcoming of this approach is the requirement in practice of large example sets to achieve good reconstructions.

Our approach on the other hand builds an illumination varying subspace by constructing artificially illuminated color images from an original image [4]. This normally requires availability of surface gradient information, which in our case may be easily computed from depth data. Since it is impossible to simulate all types of illumination conditions, we try to simulate those conditions that have the greatest effect in face recognition performance. Heterogeneous shading of the face caused by a directional light coming from one side of the face was experimentally shown to be most commonly liable for misclassification. The robustness of lighting conditions is achieved using preprocessing chain.

3.1.1 Pre-processing Chain

Pre-processing is an operation with images at the lowest level of abstraction, both input and output are intensity images. Image pre-processing methods generally involve the use of considerable redundancy in images. Neighboring pixels corresponding to one project in real images have essentially the same or similar brightness value, so if a distorted pixel can be picked out from the image. It can usually be restored as an average value of neighboring pixels [4]. Pre-processing methods can be categorized into three groups depending upon the use of a priori information:

- No knowledge about the nature of the degradation is used, only very general properties of the degradation are assumed.
- Knowledge about the properties of the image acquisition device and the condition under which the image was obtained are employed, the nature of noise is sometimes known.
- Knowledge about the objects that are searched for in the image is used to guide and simplify the pre-processing process.

Pre-processing is done to remove unwanted illumination effects, like Non-uniform illumination, Shadowing, aliasing and blurring, noise. It retains useful information like Facial features: eyes, nose, Ridges, wrinkles, local shadowing, and shading. It involves the following steps,
Step 1: Gamma correction.
Step 2: Difference of Gaussian filtering.
Step 3: Mask unwanted facial regions like beard, hair,etc…
Step 4: Contrast normalization.

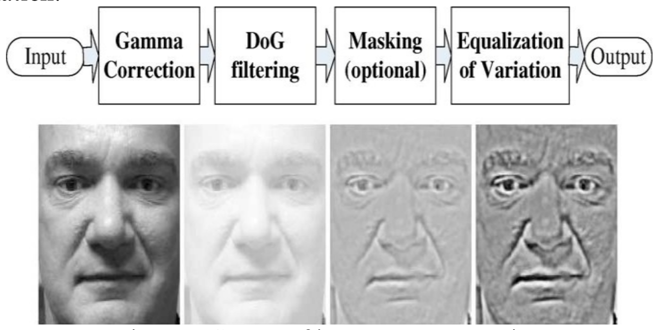


Fig. 3.1. Stages of image pre-processing

This is a preprocessing chain run before feature extraction that incorporates a series of stages designed to counter the effects of illumination variations, local shadowing, and highlights while preserving the essential elements of visual appearance.

3.1.1.1 Gamma correction

Gamma is one of the most misunderstood concepts in computer imaging applications. Part of the reason is that the term is used to mean several different things, completely confusing the user. The purpose of this memo is to draw distinctions between the several uses and to propose a fixed meaning for the term: Gamma is the term used to describe the nonlinearity of a display monitor.

All computer display monitors (and all TV sets) are nonlinear [5]. This means that if the voltage on their electronics is doubled, their brightness does not double as expected to. In fact, it varies approximately as the square of the voltage. If it varied by the square exactly, we would say it had a Gamma of 2. In other words, Gamma is an exponent. But all monitors are slightly different, so the actual Gamma of a monitor might be anywhere from 1.4 to 2.6, instead of 2. Very typical Gamma numbers are 1.8 for the PC and 2.2 for the broadcast TV but these should not be taken as gospel. They vary from display to display, even on displays from the same manufacturer. Any Gamma other than 1 is nonlinear.

Most computer displays are still based on CRTs. So the historical meaning of Gamma is that it is a number indicating the degree of nonlinearity of a given CRT. Gamma is a measure of the nonlinearity of a display device. In practice a full log transformation is often too strong, tending to over-amplify the noise in dark regions of the image, but a power law with exponent in the range [0 to 0.5] is a good compromise. Here we use Gamma = 0:2 as the default setting.

3.1.1.2 Dog Filtering

The Difference of Gaussian (DoG) module is a filter that identifies edges. The DoG filter is similar to the LOG and DoB filters in that it is a two stage edge detection process. The DoG performs edge detection by performing a Gaussian blur on an image at a specified theta (also known as sigma or standard deviation). The resulting image is a blurred version of the source image. The module then performs another blur with a sharper theta that blurs the image less than previously. The final image is then calculated by replacing each pixel with the difference between the two blurred images and detecting when the values cross zero, i.e. negative becomes positive and vice versa. The resulting zero crossings will be focused at edges or areas of pixels that have some variation in their surrounding neighborhood.

Difference of Gaussians is a gray-scale image enhancement algorithm that involves the subtraction of one blurred version of an original gray-scale image from another, less blurred version of the original [5]. The blurred images are obtained by convolving the original gray-scale image with Gaussian kernels having differing standard deviations. Blurring an image using Gaussian kernel suppresses only high-frequency spatial information. Subtracting one image from the other preserves spatial information that lies between the range of frequencies that are preserved in the two blurred images. Thus, the difference of Gaussians is similar to a band-pass filter that discards all but a handful of spatial frequencies that are present in the original gray-scale image.

Gamma correction does not remove the influence of overall intensity gradients such as shading effects. Algorithm removes high frequency detail that often includes random noise, rendering this approach one of the most suitable for processing images with a high degree of noise. When utilized for image enhancement, the Difference of Gaussians algorithm is typically applied when the size ratio of kernel (2) to kernel (1) is 4:1 or 5:1.

3.1.1.3 Masking

If a mask is needed to suppress facial regions that are felt to be irrelevant or too variable, it should be applied at this point. Otherwise, either strong artificial gray-level edges are introduced into the convolution, or invisible regions are taken into account during contrast equalization. The images before and after pre-processing are compared as follows,

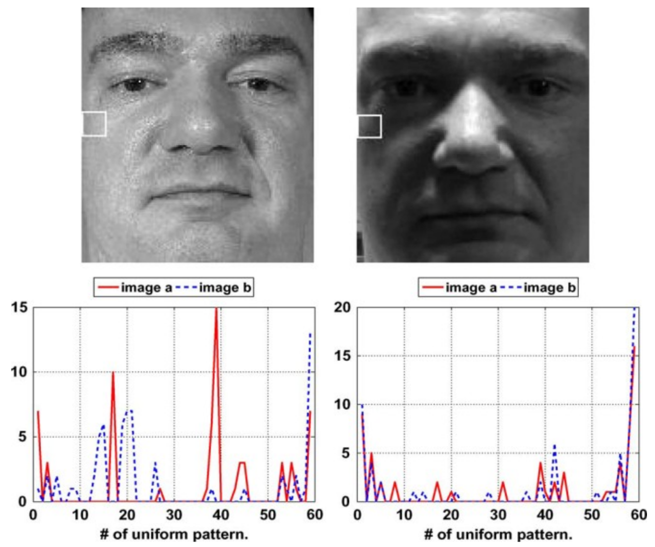


Fig. 3.2. Before pre-processing and after pre-processing

3.1.1.4 Equalization Of Variation

The final stage of our preprocessing chain rescales the image intensities to standardize a robust measure of overall contrast or intensity variation [7]. It is important to use a robust estimator because the signal typically contains extreme values produced by highlights, small dark regions such as nostrils, garbage at the image borders, etc. One could use the median of the absolute value of the signal for this, but here we have preferred a simple and rapid approximation based on a two stage process as follows:

$$I(x,y) \longrightarrow \frac{I(x,y)}{(\text{mean}(|I(x^{l},y^{l})|^{a}))^{1/a}}$$

$$I(x,y) \longrightarrow \frac{I(x,y)}{(\text{mean}(\min(r,|I(x^{l},y^{l})|^{a})))^{1/a}}$$

Here, α is a strongly compressive exponent that reduces the influence of large values, r is a threshold used to truncate large values after the first phase of normalization, and the mean is over the whole (unmasked part of the) image. By default we use α = 0.1 and r = 10 .The resulting image is well scaled but it can still contain extreme values. To reduce their influence on subsequent stages of processing, we apply a final nonlinear mapping to compress over-large values.

3.2 Robust Feature Extraction

Robustness is the quality of being able to withstand stresses, pressures, or changes in procedure or circumstance [7]. A system, organism or design may be said to be "robust" if it is capable of copying well with variations (sometimes unpredictable variations) in its operating environment with minimal damage, alteration or loss of functionality. Current feature sets offer quite good performance under illumination variations but there is still room for improvement. For example, LBP features are known to be sensitive to noise in near-uniform image regions such as cheeks and foreheads. We introduce a generalization of LBP called Local Ternary Patterns (LTPs) that is more discriminate and less sensitive to noise in uniform regions. LBP based representations typically subdivide the face into a regular grid and compare histograms of LBP codes within each region.

3.2.1 Local Binary Pattern

The Local Binary Pattern (LBP) texture analysis operator is defined as a gray-scale invariant texture measure, derived from a general definition of texture in a local neighborhood. The current form of the LBP operator is quite different from its basic version: the original definition is extended to arbitrary circular neighborhoods, and a number of extensions have been developed. The basic idea is however the same: a binary code that describes the local texture pattern is built by thresholding a neighborhood by the gray value of its center.

Through its recent extensions, the LBP operator has been made into a really powerful measure of image texture, showing excellent results in terms of accuracy and computational complexity in many empirical studies. The LBP operator can be seen as a unifying approach to the traditionally divergent statistical and structural models of texture analysis.

The LBP feature vector, in its simplest form, is created in the following manner:

- Divide the examined window to cells (e.g. 16x16 pixels for each cell).
- For each pixel in a cell, compare the pixel to each of its 8 neighbors (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels along a circle, i.e. clockwise or counter-clockwise.
- Where the center pixel's value is greater than the neighbor, write "1". Otherwise, write "0". This gives an 8-digit binary number (which is usually converted to decimal for convenience)
- Compute the histogram, over the cell, of the frequency of each "number" occurring (i.e., each combination of which pixels are smaller and which are greater than the center).
- Optionally normalize the histogram.
- Concatenate normalized histograms of all cells. This gives the feature vector for the window.

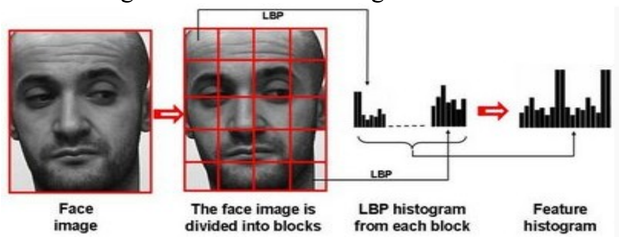

LBP
LBP
Face image
The face image is divided into blocks
LBP histogram from each block
Feature histogram

Fig. 3.3. Face description with LBP

This histogram effectively has a description of the face on three different levels of locality: the LBP labels for the histogram contain information about the patterns on a pixel-level, the labels are summed over a small region to produce information on a regional level and the regional histograms are concatenated to build a global description of the face.

The LBP operator takes a local neighborhood around each pixel, thresholds the pixels of the neighborhood at the value of the central pixel and uses the resulting binary-valued image patch as a local image descriptor. It was originally defined for 3x 3 neighborhoods, giving 8-bit integer LBP codes based on the eight pixels around the central one [8]. Formally, the LBP operator takes the form,

$$\mathrm{LBP}\,(x_c, y_c) = \sum_{n=0}^{7} 2^n \, s(i_n - i_c)$$

where in this case n runs over the 8 neighbors of the central pixel c.

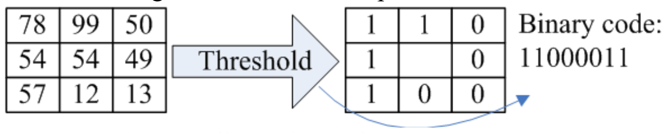


Fig. 3.4. Illustration of basic LBP operator

Two extensions of the original operator were made in LBP. The first defined LBPs for neighborhoods of different sizes, thus making it feasible to deal with textures at different scales. The second defined the so-called uniform patterns: an LBP is "uniform" if it contains at most one 0-1 and one 1-0 transition when viewed as a circular bit string. For example, the LBP code in it is uniform. Uniformity is important because it characterizes the patches that contain primitive structural information such as edges and corners. Although only 58 of the 256 8-bit patterns are uniform, nearly 90% of all observed image neighborhoods are uniform and many of the remaining ones contain essentially noise. Thus, when histogramming LBPs the number of bins can be reduced significantly by assigning all nonuniform patterns to a single bin, typically without losing too much information.

3.2.2 Local Ternary Patterns

LTP consistently outperforms LBP by a small Margin, and the DT based similarity metric significantly improves on histogram distance independent of the local region size [8]. By way of comparison, the best previous performance that we are aware of on this dataset is 78.0%.

LBPs have proven to be highly discriminative features for texture classification and they are resistant to lighting effects in the sense that they are invariant to monotonic gray-level transformations. However because they threshold at exactly the value of the central pixel they tend to be sensitive to noise, particularly in near-uniform image regions, and to smooth weak illumination gradients. This section extends LBP to 3-valued codes, LTP, in which gray-levels in a zone of width around are quantized to zero, ones above this are quantized to 1 and ones below it to 1, i.e., the indicator is replaced with a 3-valued function and the binary LBP code is replaced by a ternary LTP code. Here is a user-specified threshold so LTP codes are more resistant to noise, but no longer strictly invariant to gray-level transformations.

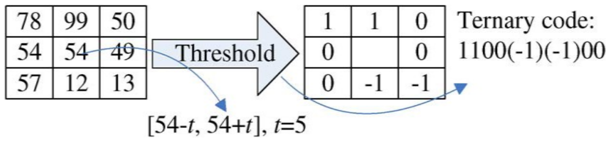


Fig. 3.5. Illustration Of Basic LTP Operator

The LTP transformation is given by,

$$s'(u, ic, t) = \begin{cases} 1, & u >= ic + t \\ 0, & |u - ic| < t \\ -1, & u <= ic - t \end{cases}$$

Here the threshold was set to 5. When using LTP for visual matching, we could use valued codes, but the uniform pattern argument also applies in the ternary case. For simplicity, the experiments below use a coding scheme that splits each ternary pattern into its positive and negative halves subsequently treating these as two separate channels of LBP descriptors for which separate histograms and similarity metrics are computed, combining the results only at the end of the computation.

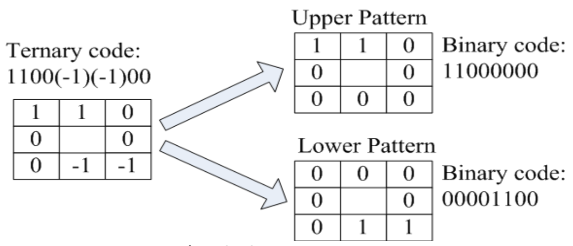
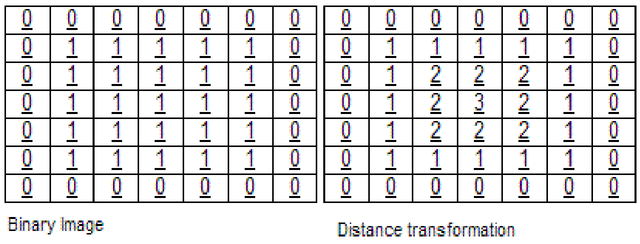


Fig. 3.6. LTP To LBP

When using LTP for visual matching, we could use valued codes, but the uniform pattern argument also applies in the ternary case.

3.2.3 Distance Transform Similarity Metric

A distance transform, also known as distance map or distance field, is a derived representation of a digital image. The choice of the term depends on the point of view on the object in question: whether the initial image is transformed into another representation, or it is simply endowed with an additional map or field. The map labels each pixel of the image with the distance to the nearest obstacle pixel.



0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

Binary Image

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	2	2	2	1	0
0	1	2	3	2	1	0
0	1	2	2	2	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

Distance transformation

Fig. 3.7. Distance Transform

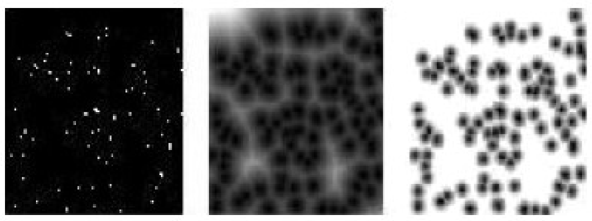
To apply the distance transform similarity metric for a reference image X and a test image Y, first their LBP/LTP codes must be found. Then transform these into a set of sparse binary images, one for each code and after that calculate the distance transform image of each binary reference image. For each test image position, compare it with the corresponding reference distance.

Distance calculation can be done by using the formula,

$$D(X,Y) = \sum_{\text{pixels } (i,j) \text{ of } Y} w\left(d_X^{k_Y(i,j)}(i,j)\right)$$

where

- Each pixel of d_x gives the distance to the nearest image X pixel with code k.
- kY(i,j)is the code value of pixel (i,j) of image Y.
- W is a monotonically increasing function.



Figu. 3.8. Binary Layer, After Distance Transform and Truncated Version

The above figure explains the binary later of an image, its distance transform and the truncated linear version of the image. This transform is also used in motion planning in robotics, and even path finding.

3.3 Phase Congruency

Phase congruency reflects the behavior of the image in the frequency domain. Gradient-based operators, which look for points of maximum intensity gradient, will fail to correctly detect and localize a large proportion of features within images [9]. Unlike the edge detectors which identify the sharp changes in intensity, the phase congruency model detects points of order in the phase spectrum.

3.3.1 Algorithmic Steps

1. Normalize all the training images to a size of 64 x 64
2. Calculate the phase congruency at each pixel position for each training image

3. Modularize the 64 x 64 dimension phase congruency maps into non-overlapping regions of size 4 x 4
4. Concatenate two horizontally adjacent modules of size 4 x 4 to form a 4 x 8 pixel block called concatenated horizontal block (CHB)
5. Create combinations of two CHB's within the defined neighborhood of 8 CHBs to form a set of 28 different modules of size 8 x 8 pixels each
6. Index the modules (8x8) according to the location of the two CHBs involved in the CHB

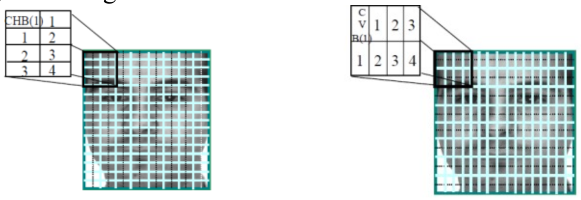


Figure 3.9. Creation of CHB and CVB

7. Concatenate two 4 x 4 blocks into concatenated vertical blocks (CVB) of size 8 x 8 modules
8. Follow the procedure from step 4 to step 6 in creating 8 x 8 modules
9. Combine the set of modules formed in step 5 and 8. Also eliminate the redundant modules
10. Arrange the modules according to the increasing order of indexing given within the neighborhood
11. Take one module at a time consecutively from each neighborhood block to make a set of 64 modules
12. Create eigen subspaces for each of the modules that are created after arrangement
13. Classify the probe image based on voting procedure to create the output image

IV. RESUTS AND COMPARISON

The existing systems such as Eigen faces, Gabor Wavelets, Neural Networks and Genetic Algorithm are difficult with range of appearance variations that occur in unconstrained natural images due to light, pose, facial expressions. Our project mainly focuses on the issue of robustness to lighting variations and also it helps recognition even though there are expression variations.

Run time is also a critical factor in many applications. Our method uses only simple closed-form image operations so it is much more efficient than ones that require expensive iterative optimizations such as Logarithmic Total Variation (LTV) and anisotropic diffusion. Our (MATLAB) implementation takes only about 50 ms to process a 128 x 128 pixel face image on a 2.8-GHz, allowing face preprocessing to be performed in real time and thus providing the ability to handle large face databases. In comparison, the current implementation of Gross and Brajovic's anisotropic smoothing (GB) is about 5 times slower and LTV is about 300 times slower.

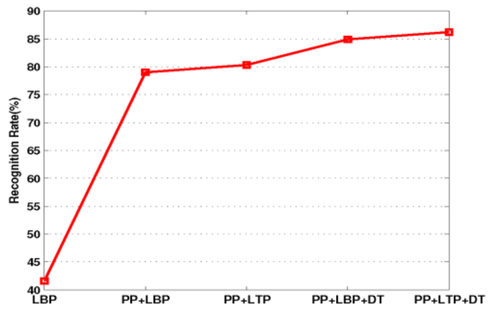


Fig. 4.1. Plot of Percentage of Recognition Rate

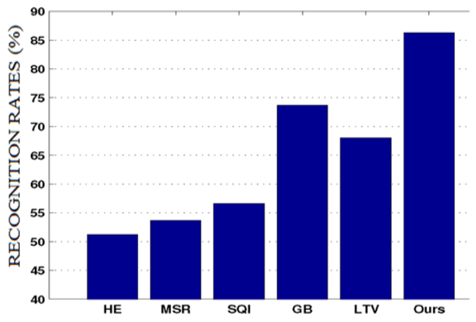
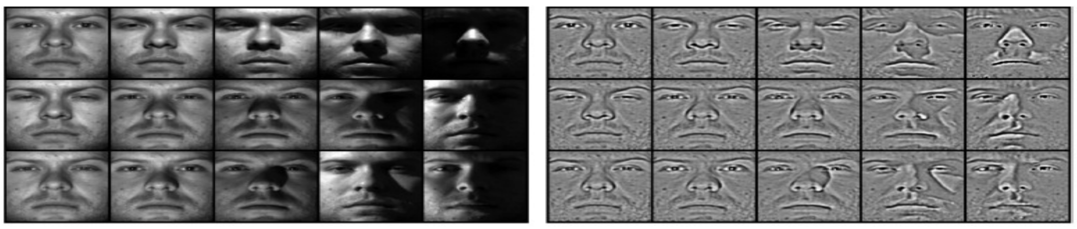


Fig. 4.2. Comparison of Recognition Rates

PP – PREPROCESSING
LBP – LOCAL BINARY PATTERN
LTP – LOCAL TERNARY PATTERN
DT – DISTANCE TRANSFORM

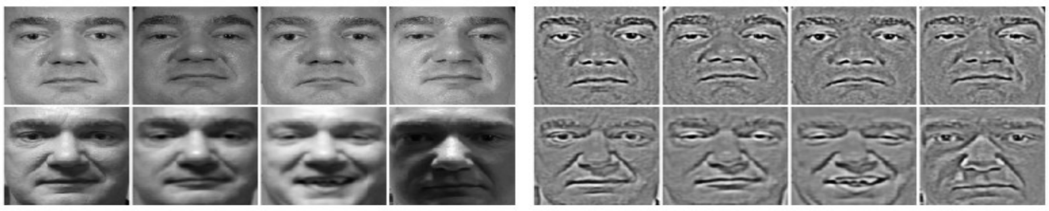
This paper is able to achieve efficiency about 88.1%. Under difficult lighting conditions are overcome effectively using a combination of preprocessing chain, LBP, LTP and distance transform similarity metric.

The effectiveness of our methods can be illustrated by presenting experiments on three large-scale face data sets with difficult lighting conditions: Extended Yale B, CAS-PEAL-R1, and Face Recognition Grand Challenge (FRGC) version 2 Experiment 4. They are the database containing faces under different illumination conditions. The effectiveness of their faces before and after normalization are given as follows,



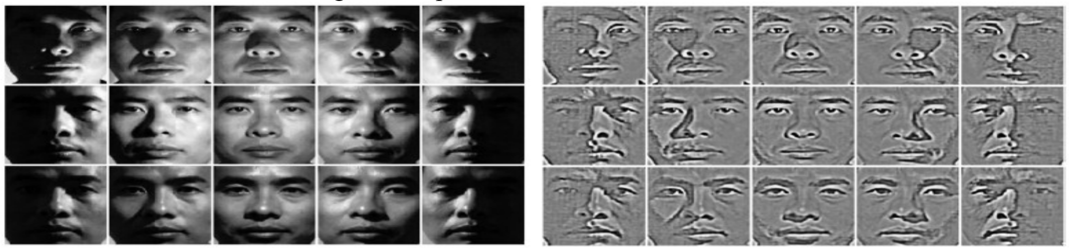
Before normalization After normalization

Fig. 4.3. Experiments on Extended Yale B



Before normalization After normalization

Fig. 4.4. Experiments on FRGC Version 2



Before normalization After normalization

Fig. 4.5. Experiments on CAS-PEAL-R1

V. CONCLUSION

We have presented new methods for face recognition under uncontrolled lighting based on robust preprocessing and an extension of the LBP. The main contributions are as follows:

- A simple, efficient image preprocessing chain whose practical recognition performance is comparable to or better than current illumination normalization methods.
- A rich descriptor for local texture called LTP that generalizes LBP while fragmenting less under noise in uniform regions.
- A distance transform based similarity metric that captures the local structure and geometric variations of LBP/LTP face images better than the simple grids of histograms that are currently used.
- A heterogeneous feature fusion-based recognition framework that combines two popular feature sets such as Gabor wavelets and LBP with robust illumination normalization and a kernelized discriminative feature extraction method. Moreover, we empirically make a comprehensive analysis and comparison with several illumination normalization methods on the large-scale FRGC-204 dataset, and investigate their connections with robust descriptors, recognition methods, and image quality.

5.1 Future Enhancements

Face detection and verification can be done not only with the still image or raw image. It also can be done with the video image. Our project can detect the face from a video. To detect the image which is in the video, the preprocessing chain is used. But the process in the pre-processing chain has to be changed. Face detection and verification is used in access control, visitor management system, time attendance system.

REFERENCES

[1] Adini. Y, Moses. Y, and Ullman. S, (Jul. 1997) "Face recognition: The problem of compensating for changes in illumination direction," IEEE Trans.Pattern Anal. Mach. Intel., vol. 19, no. 7, pp. 721–732.

[2] Ahonen. T, Hadid. A and Pietikainen. M, (Dec. 2006) "Face description with local binary patterns: Application to face recognition," IEEE Trans.pattern Anal. Mach. Intel., vol. 28, no. 12, pp. 2037– 2041.

[3] Dalal. N and Trigg's N, (2005) "Histograms of oriented gradients for human detection," in Proc. CVPR, Washington, DC, pp. 886–893.

[4] Lee. K, Ho. J, and Kriegman. D, (May 2005) "Acquiring linear subspaces for face recognition under variable lighting," IEEE Trans. Pattern Anal.

[5] Ojala. T, Pietikainen. M and Maenpaa. T, (Jul. 2002) "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,"IEEE Trans. Pattern Anal. Mach. Intel., vol. 24, no. 7, pp. 971–987.
[6] Rafael C. Gonzalez and Richard E Woods,(2005)"Digital Image Processing", Person Education Asia.
[7] Shan. S, Gao, Cao. B, and Zhao. D, (2003) "Illumination normalization for robust face recognition against varying lighting conditions," in Proc. AMFG, Washington, DC, p. 157.
[8] Tan. X and Triggs. B, (2007) "Enhanced local texture feature sets for face recognition under difficult lighting conditions," in Proc. AMFG, pp. 168–182.
[9] Satyanadh Gundimada and Vijayan K Asari " A Novel Neighborhood Defined Feature Selection on Phase Congruency Images for Recognition of Faces with Extreme Variations" International Journal of Information Technology Volume 3 Number 1.