# Eigenimage Based Pain Expression Recognition

Md. Maruf Monwar*, Siamak Rezaei* and Dr. Ken Prkachin

**Member, IEEE*

Abstract — **In this paper, we present a robust approach for pain expression recognition from video sequences. An automatic face detector is employed which uses skin color modeling to detect human face in the video sequence. The pain affected portions of the face are obtained by using a mask image. The obtained face images are then projected onto a feature space, defined by Eigenfaces, to produce the biometric template. Pain recognition is performed by projecting a new image onto the feature spaces spanned by the Eigenfaces and then classifying the painful face by comparing its position in the feature spaces with the positions of known individuals.**

***Index Terms*— Chromatic color space, Human face detection, Skin color model, Eigenface, Pain recognition.**

## I. Introduction

In recent years, tremendous amount of researches have been carried out in the field of automatic expressions (such as, pain, anger, sadness etc.) recognition from video sequence and still has significant potential for further research and development. This coupled with its vast array of commercial applications (like in medical system, in psychological research etc.) make it an attractive area of research. In this paper, we propose a method for automatically inferring pain in video sequences.

We have focused our research towards developing an unsupervised pain recognition scheme that does not depend on excessive geometry and computations like deformable templates. Instead, Eigenface approach is used. In this approach, Eigenfaces are obtained by performing Principal Component Analysis (PCA) technique on a set of facial images. This approach seems to be an adequate method to be used in the pain recognition due to its simplicity, speed and learning capability [1]. Although pain recognition is a fundamental step in a fully automated facial expression analysis system, the first important step in pain recognition is detecting faces in video sequences which requires to know where the skin regions are. This is called skin region detection. Most of the existing expression recognition methods use gray intensity values to detect faces [2]-[3] in a video stream. However, it is a well-known fact that the majority of images acquired today are colored and the skin color features should be important sources of information for discriminating faces from the background. In this system, color is modeled as a Gaussian function in chromatic color space [4], where intensity plays no role and whole information is provided by hue, saturation in other word in pure color (r, g). Each detected skin regions are then tested for the presence of face in the next steps. Approximate face locations are detected using a proper height-width proportion of general face. Once rough face locations are detected, they are verified by an eye template-matching scheme. There is biological evidence that eyes play the most important role in human face detection [5]-[6]. Once a face is detected, the pupils, nostrils and lip corners are located and face is cropped from the image. Then face images are projected onto a feature space to produce the biometric template. The feature space is defined by the Eigenfaces [7], which are eigenvectors for a set of faces.

Recognition is performed by projecting a new image onto the feature spaces spanned by the Eigenfaces and then classifying the painful face by comparing its position in the feature spaces with the positions of known individuals. Actual system is capable of both recognizing individuals having pain and learning to recognize new face images with pain. The Eigenface approach used in this scheme has advantages over other expression recognition methods in its speed, simplicity, learning capability and robustness to small changes in the face image.

## II. Skin color modeling for face detection

### A. Image Acquisition

We have used a database of painful and normal video files. In this database, there are two video files for every subject and a total of 38 subjects with different colors, ethnicities, ages and genders are considered. In one file, the subject is in normal mood and in the other file, the subject is in painful mood due to moving hands or pressing something or shaking heads etc. The resolutions of the videos are 96 x 96. These video files are first read and the numbers of frames of each video are determined. Middle frame of the videos are then stored as image in the database for further processing. The reason for taking middle frame is that, in almost all the pain videos, the expression for pain begins after some time from the starting of the videos and

Manuscript received March 6, 2006. This work has been done as a requirement of a part of an M.Sc thesis.

Md. Maruf Monwar is from the University of Northern British Columbia, Prince George, Canada. He is a graduate student of Computer Science. He is a member of IAENG and a student member of IEEE.
(phone: 1-250-612-0859; fax: 1-250-960-5544; e-mail: monwar@unbc.ca ).

Siamak Rezaei is an Associate Professor of Computer Science of the University of Northern British Columbia, Prince George, Canada.
(phone: 1-250-960-6263; fax: 1-250-960-5544; e-mail: siamak@unbc.ca).

Dr. Ken Prkachin is a professor of Psychology department of the University of Northern British Columbia, Prince George, Canada.
(phone: 1-250-960-6633; fax: 1-250-960-5744; e-mail: kmprk@unbc.ca).

ends some time before the ending of the videos. So, by taking the middle frame, it is ensured that the expression for pain in a pain video will be captured. The videos are roughly 1 to 1.5 second long

### B. *Skin Color Model*

In order to segment human skin regions from non-skin regions based on color, we need a reliable skin color model that is adaptable to people of different skin colors and to different lighting conditions. The common RGB representation of color images is not suitable for characterizing skin-color. In the RGB (red, green and blue) space, the triple component (r, g, b) represents not only color but also luminance. Luminance may vary across a person's face due to the ambient lighting and is not a reliable measure in separating skin from non-skin region [10]. Luminance can be removed from the color representation in the chromatic color space. Chromatic colors [11], also known as "pure" colors in the absence of luminance, are defined by a normalization process shown below:

$$r = R/(R+G+B)$$

$$b = B/(R+G+B)$$

Color green is redundant after the normalization because r + g + b = 1. If two points $P_1[r_1,g_1,b_1]$ and $P_2[r_2,g_2,b_2]$, are proportional, i.e.,

$$\frac{r_1}{r_2} = \frac{g_1}{g_2} = \frac{b_1}{b_2}$$

then, $P_1$ and $P_2$ have the same color but different brightness.

Chromatic colors have been effectively used to segment color images in many applications. It is also well suited in this case to segment skin regions from non-skin regions. The color distribution of skin colors of different people was found to be clustered in a small area of the chromatic color space. Skin colors of different people are very close, but they differ mainly in intensities [10]. With this finding, we could proceed to develop a skin-color model in the chromatic color space. Figure 1(a) and 1(b) illustrate the training process, in which a skin-color region is selected and its RGB representation is stored. It was verified, using training data, that skin-colors are clustered in color space, as illustrated in Fig. 2(a). Although skin colors of different people appear to vary over a wide range, they differ much less in color than in brightness. In other words, skin colors of different people are very close, but they differ mainly in intensities [8]. With this finding, we could proceed to develop a skin-color model in the chromatic color space.

A total of 68 skin samples from 68 color images taken from the same number of videos (normal and painful) were used to determine the color distribution of human skin in chromatic color space and generate the statistical skin-color model. Our samples were taken from persons of different ethnicities: Asian, Caucasian and African and from different ages and genders with varying illumination condition. As the skin samples were extracted from color images, the skin samples were filtered using a low-pass filter to reduce the effect of noise in the samples.

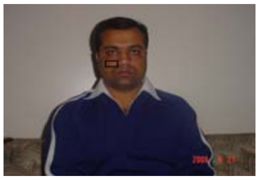

Fig. 1(a). Selected skin region in RGB image

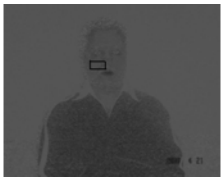

Fig. 1(b). Selected skin in Chromatic color

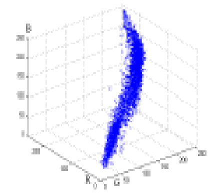

Fig. 2(a). Cluster in color space (RGB)

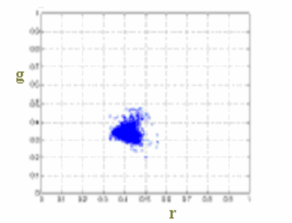

Fig. 2(b). Cluster in chromatic space (r, g)

The color histogram revealed that the distribution of skin-color of different people are clustered in the chromatic color space and a skin color distribution can be represented by a Gaussian model N(m, C) [4], where:

Mean, m = E {x} [ where $x = (r\ \ b)^T$ ] and

$$\text{Covariance, } \sum = \begin{bmatrix} \sigma_{rr} & \sigma_{rg} \\ \sigma_{gr} & \sigma_{gg} \end{bmatrix}$$

With this Gaussian fitted skin color model, we can now obtain the likelihood of skin for any pixel of an image. Therefore, if a pixel, having transform from RGB color space to chromatic color space, has a chromatic pair value of (r,b), the likelihood of skin for this pixel can then be computed as follows:

$$\text{Likelihood} = P(r,b) = \exp[-0.5(x-m)^T C^{-1}(x-m)],$$
$$[\text{ where, } x = (r,b)^T\ ]$$

Hence, this skin color model can transform a color image into a gray scale image such that the gray value at each pixel shows the likelihood of the pixel belonging to the skin. With appropriate thresholding, the gray scale images can then be further transformed to binary images showing skin regions and non-skin regions.

### C. *Skin Region Segmentation*

Our main goal in this segmentation process is to remove the background of the image from skin regions using previously discussed skin color model. First, input image is converted to chromatic color space. Using Gaussian model, a grayscale image of skin likelihood pixels is constructed and skin pixels have some set of constant values for each r, g and b component. Every pixel in normalized image has three values and they are normalized-red, normalized-green and normalized-blue.

Segmentation process extracts these normalized components and constructs two images (fig. 4(c) and fig. 4(d)). Each of these images is converted into black and white image by applying different threshold for normalized input image (fig. 4(e) and fig. 4(f)) such that, r = 0.41-0.50 and g = 0.21- 0.30. Finally, we perform an ‘AND’ operation between these two black and white images where white pixels are skin and blacks are non skin pixel. In this approach, due to noise and distortion in input image, color information of some skin pixels acts like non skin region and generate non contiguous skin color region. To solve this problem, first morphological closing operator is used to obtain skin-color blobs (fig. 4(g)). A median filter was also used to eliminate spurious pixels (fig. 4(h)). Boundaries of skin-color regions are determined using a region growing algorithm in the binary image. Regions with size less than 1% of image size are eliminated [5]. At the end of the segmentation process, black and white skin regions of images are multiplied by the original RGB image and we then get the skin region (fig. 4(i)) of face. Fig. 3 illustrates a simple block diagram for the segmentation process and fig. 4 shows an example of segmentation and face location detection process performed on a painful image.

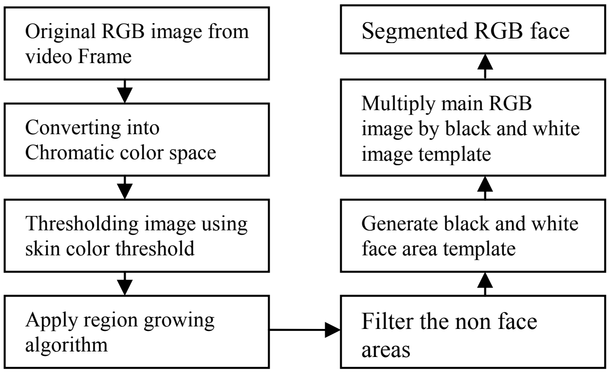
Original RGB image from video Frame
Converting into Chromatic color space
Thresholding image using skin color threshold
Apply region growing algorithm
Filter the non face areas
Generate black and white face area template
Multiply main RGB image by black and white image template
Segmented RGB face

Fig. 3. Block diagram for face segmentation

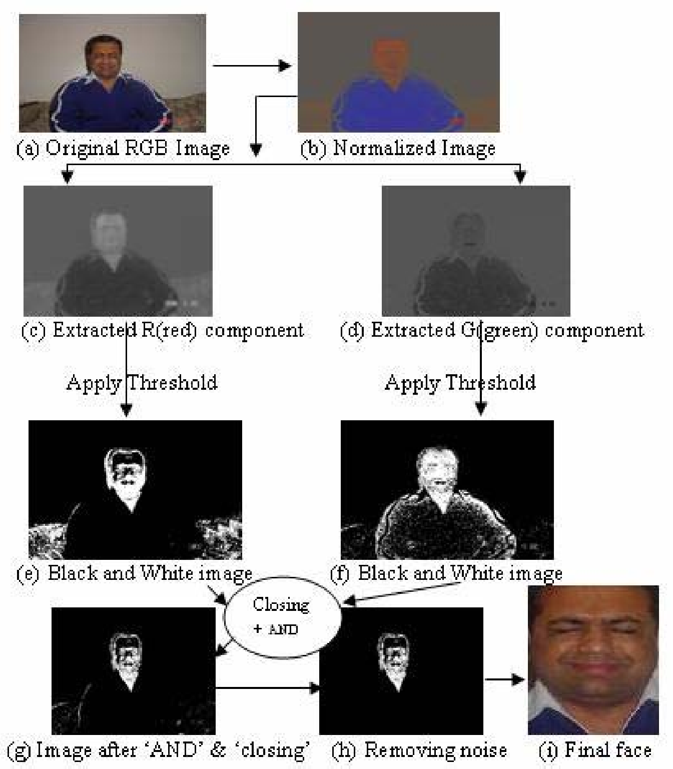
(a) Original RGB Image
(b) Normalized Image
(c) Extracted R(red) component
(d) Extracted G(green) component
Apply Threshold
Apply Threshold
(e) Black and White image
(f) Black and White image
Closing + AND
(g) Image after ‘AND’ & ‘closing’
(h) Removing noise
(i) Final face

Fig. 4. Segmentation and approximate face location process

### D. Face Detection

To reduce some search space for eyes template matching, bounding rectangles of all connected areas from black-white template are taken into consideration and the center of the face areas will be calculated. This is the mass point of the template area. Now the calculation of the height and width of the bounding rectangle can be performed. If the height-width proportions satisfy for face like shape, we keep those areas, otherwise we remove these. Thus the template with the approximate face area is multiplied by the original image and we get the face. Then to consider only the meaningful portions of the face we use a mask image. A bitwise ‘AND’ operation is used to apply the mask image with the original face image. Features in the image which coincide with the white areas on the mask image will be displayed.

The original video frame, the obtained gray level image, the mask image and the resultant image is shown in Fig. 5.

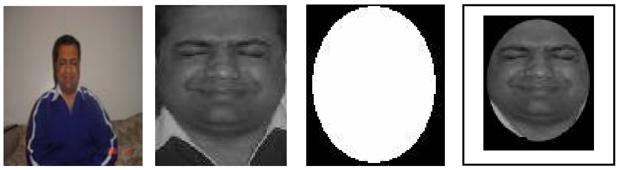

Fig. 5(a) Fig. 5(b) Fig. 5(c) Fig. 5(d)

Fig. 5. (a) Original video frame (b) Gray level image (c) Mask image and (d) Resultant image

.

## III. PAIN RECOGNIION USING EIGENFACES

### A. Eigenface Method

The motivation behind Eigenfaces is that the previous works ignore the question of which features are important for the classification, and which are not. Eigenfaces approach seeks to answer this by using Principal Component Analysis (PCA) of the facial images. This analysis reduces the dimensionality of the training set, leaving only those features that are critical for face recognition.

*1) Defining Eigenfaces:* The Eigenfaces method looks at the face as a whole. In this method, a collection of face images is used to generate a two-dimensional gray-scale image to produce the biometric template. In our proposed system, first the face images are processed by the face detector. These processed images are used as the training set for the Eigenfaces method. Then we calculate the Eigenfaces from the training set, keeping only the M Eigenfaces which correspond to the highest Eigenvalues. These M images denote the "face space" [7]. Finally we calculate the corresponding location or distribution in M-dimensional weight space for each known individual, by projecting their face images (from the training set) onto the "face space".

When a new face image is encountered, we calculate a set of weights based on the input image and the M Eigenfaces by projecting the input image onto each of the Eigenfaces. We determine if the image is a face (known or unknown) by checking to see if the image is sufficiently close to "face space"— i.e. determining the ability of the Eigenfaces to reconstruct the image. If it is a face, we classify the weight pattern as either a known person or as unknown person.

*2) Calculating Eigenfaces:* Let a face image I(x, y) be a two-dimensional N X N array of (8-bit) intensity values. Such an image may also be considered as a vector of dimension N$^2$, so that a typical image of size N X N becomes a vector of dimension N$^2$ or, equivalently, a point in N$^2$-dimensional space. Images of faces, being similar in overall conjuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The main idea of the PCA method is to find the vectors which best account for the distribution of face images within the entire image space. These vectors define the subspace of face images called "face space". Each vector is of length N$^2$, describes an N X N image, and is a linear combination of the original face images. Because these vectors are the eigenvectors of the covariance matrix corresponding to the original face images, and because they are face-like in appearance, they are referred to as Eigenfaces [7].

Steps for Eigenfaces calculation:

1. The first step is to obtain a set S with M face images. Each image is transformed into a vector of size N and placed into the set.

$$S = \{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \cdots\cdots\cdots, \Gamma_M\}$$

2. Second step is to obtain the mean image Ψ.

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$$

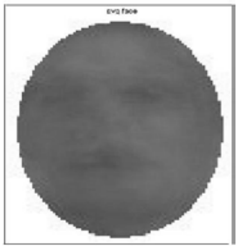

Fig. 6. Average face image

3. Then we find the difference Φ between the input image and the mean image

$$\Phi_i = \Gamma_i - \Psi$$

4. Next we seek a set of M orthonormal vectors, u$_n$, which best describes the distribution of the data. The k$_{th}$ vector, u$_k$, is chosen such that

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}\left(u_k^T\Phi_n\right)^2$$

5. $\lambda_k$ is a maximum, subject to

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1 & \text{If } l = k \\ 0 & \text{Otherwise} \end{cases}$$

where u$_k$ and λ$_k$ are the eigenvectors and Eigenvalues of the covariance matrix C

6. The covariance matrix C has been obtained in the following manner

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = AA^T \text{ [where, } A = \{\Phi_1, \Phi_2, \cdots\cdots, \Phi_n\} \text{ ]}$$

7. To find eigenvectors from the covariance matrix is a huge computational task. Since M is far less than N$^2$ by N$^2$, we can construct the M by M matrix,

$$L = A^TA, \quad [\text{ where } L_{mn} = \Phi_m^2\Phi_n \text{ ]}$$

8. We find the M Eigenvectors, v$_l$ of L.
9. These vectors (v$_l$) determine linear combinations of the M training set face images to form the Eigenfaces u$_l$

$$u_l = \sum_{k=1}^{M} v_{lk}\Phi_k \qquad l = 1, 2, \cdots\cdots\cdots, M$$

After computing the Eigenvectors and Eigenvalues on the covariance matrix of the training images, these M eigenvectors are sorted in order of descending Eigenvalues and chosen to represent Eigenspace. Fig. 7 shows the eigenfaces of the training face images:

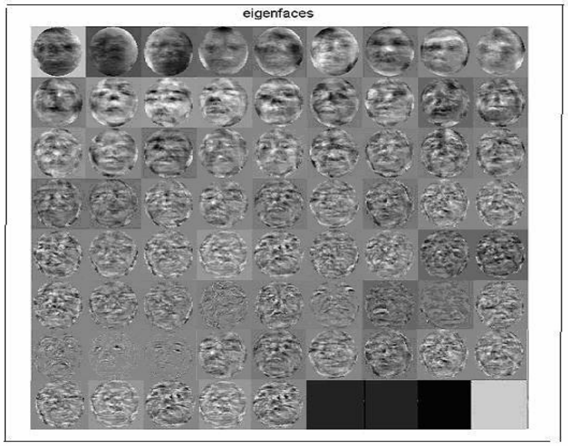

Fig. 7. Eigenfaces for the training image set

Finally, we project each of the original images into Eigenspace. This gives a vector of weights representing the contribution of each Eigenface to the reconstruction of the given image.

*3) Recognition using Eigenfaces:* Once Eigenspace has been defined, we can project any image into Eigenspace by a simple matrix multiplication:

$$\omega_k = u_k^T(\Gamma - \Psi) \quad \textit{and } \Omega^T = \left\lceil \omega_1, \omega_2, \cdots\cdots\cdots, \omega_M \right\rceil$$

where, $u_k$ is the k^th^ eigenvector and $\omega_k$ is the k^th^ weight in the vector $\mathbf{\Omega}^T = [\omega_1, \omega_2, \omega_3, \ldots \omega_M]$. The M weights represent the contribution of each respective Eigenfaces. The vector $\mathbf{\Omega}$, is taken as the 'face-key' for a face's image projected into Eigenspace. We compare any two 'face-keys' by a simple Euclidean distance measure, $\epsilon = \| \mathbf{\Omega}_a - \mathbf{\Omega}_b \|^2$. An acceptance (the two face images match) or rejection (the two images do not match) is determined by applying a threshold. Any comparison producing a distance below the threshold is a match. The steps for recognition process are as follows:

1. When an unknown face is found, project it into Eigenspace.
2. Measure the Euclidean distance between the unknown face's position in Eigenspace and all the know faces' positions in Eigenspace.
3. Select the face closest in Eigenspace to the unknown face as the match.

## IV. Simulations and Results

The proposed system is applied on a wide variety of painful and normal video. The videos include different lightening conditions and with different backgrounds. It is found that the system successfully detects skin region of the images collected from video analysis. However, it is important to note that not all detected regions contain faces. Some corresponds to parts of human body, while other corresponds to objects with colors similar to those of skin. We implemented the entire algorithm in MATLAB 7.0 on a PENTIUM-IV windows XP workstation.

To examine the accuracy of our proposed pain recognition system, we have tested our system several times. 68 different videos are used for this experiment. Some videos contain the same person but in different orientation. The average skin region detection rate is almost 88% - 90% and average false skin region detection rate is almost 7% - 9%. Average face detection rate is almost 90% - 92% because there are some videos in which the background is very similar to the skin color. From the performance of the face detection subpart of the system, it can be inferred that the system is almost evenly applicable for videos of different ethnicity people and of different backgrounds.

## V. Conclusions

We have presented a pain recognition approach in this paper. For this we have addressed the problems of how to find a human face in a video sequence and how to represent and recognize pain expression presented in those faces. Skin color modeling approach is used for face detection and Eigenface is used for pain recognition in those faces. Though some false detection occurs, the overall pain recognition performance of the proposed system is still quite satisfactory. Detecting all the expressions from real time video sequences is our future work.

## Acknowledgement

This research has been conducted in the Computer Science program of UNBC, Canada. We would like to thank Professor Ken Prkachin at Department of Psychology, UNBC and his co-investigator Dr. Patty Solomon at the McMaster University, Canada (who collected the video database) for providing the video sequences for this research.

## References

[1] P. Sinha, "Object recognition via image invariants: a case study", *Investigative Ophthalmology and Visual Science*, vol. 35, 1994, pp.. 1735-1740.

[2] H. A. Rowley. S. Bluja, T. Kanade, "Neural network-based face detection", *The IEEE Transaction on Pattern. Analysis and Matching. International*, vol. 20 (1), 1998, pp. 39-51.

[3] M.C. Burl, T.K. Leung, P. Perona, "Face localization via shape statistics", *in the proceedings of the 1st International Workshop on Face and Gesture Recognition*, Zurich, Switzerland, 1995.

[4] Demir Gökalp, "Skin color based face detection", *Department of Computer Engineering, Bilkent University, Turkey*.

[5] R.S. Feris, T. E. de Campos, and R. M. C. Junior, "Detection and tracking of facial features in video sequences", *in the proceedings of the Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence*, 2000, pp. 127–135.

[6] B.Moghaddam and A. Pentland. "Face recognition using view-based and modular Eigenspaces", *Automatic Systems for the Identification and Inspection of Humans, SPIE*, 2277, July, 1994.

[7] M. A. Turk and A. P. Pentland, “Face recognition using Eigenfaces”, in *the proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1991, pp. 586-591.

[8] Jie Yang and Alex Waibel, "A real-Ttime face tracker", *in the proceedings of the 3rd IEEE Workshop on Applications of Computer Vision, WACV '96,* Sarasota, Fla. 1996.

[9] J. Cai, A. Goshtasby, C. Yu, "Detecting human faces in color images," *in the proceedings of the International Workshop on Multimedia Database Management Systems*, 1998, pp. 0124.

[10] G. Wyszecki and W.S. Styles, “Color science: concepts and methods, quantitative data and formulae”, *second edition*, John Wiley & Sons, New York, 1982.

[11] Y. Gong and M. Sakauchi, "Detection of regions matching specified chromatic features", *Computer Vision and Image Understanding*, vol. 61(2), 1995, pp. 263 – 269.