Face Recognition using a New Texture Representation of Face Images

Sanun Srisuk†, Kongnat Ratanarangsank†, Werasak Kurutach⋆ and Sahatsawat Waraklang⋆
†Advanced Machine Intelligence Research Laboratory (AMIRL)
Department of Computer Engineering
⋆Department of Information Technology
Mahanakorn University of Technology, Bangkok, THAILAND, 10530
E-mail: s.srisuk@mut.ac.th

Abstract

In this paper, we present a new texture representation of face image using a robust feature from the Trace transform. The masked Trace transform (MTT) offers "texture" information for face representation which is used to reduce the within-class variance. We first transform the image space to the Trace transform space to produce the MTT. Weighted Trace transform (WTT) identifies the tracing lines of the MTT which produce similar values irrespective of intraclass variations. A new distance measure is proposed by incorporating the WTT for measuring the dissimilarity between reference and test images. Our method is evaluated with experiments on AR face database.

1. Introduction

Face recognition is substantially different from classical pattern recognition problems, such as object recognition. The shapes of the objects are usually different in an object recognition task, while in face recognition one always identifies objects with the same basic shape. This is of utmost difficulty for a face recognition system when one tries to discriminate faces all of which have the same shape with minor texture differences. The face recognition therefore depends heavily on the particular choice of face representation. The aim of feature selection in face representation method is to suppress the variations of face images and simultaneously provide enhanced discriminatory power. It has many image representations proposed for face recognition such as eigenface and fisherface methods. The goal of the eigenface method is to linearly projecting the image space to a feature space of lower dimensionality [4]. One can reconstruct a face-like image by using only a few eigenvectors which correspond to the largest eigenvalues. Eigenface is an optimal reconstruction method in the sense of minimum mean square error, which projects the image on the directions that maximize the total scatter across all classes of face images. This means that the eigenface is not the optimal method in the sense of discrimination ability, which depends on the separation between different classes rather than the spread of all classes. For the problems of class separation, a method based on class specific linear projection was proposed by Belhumeur *et al.* [5]. This method tries to find the eigenvectors for which the ratio of the between-class scatter and the within-class scatter is maximized. In general, it is common to use principal component analysis in conjunction with linear discriminant analysis (LDA) to overcome the complication of singular eigenvectors. This is achieved by using PCA for reducing the dimensionality of the image space to obtain the eigenspace, and then applying the standard Fisher LDA (FLD) to further reduce the dimensionality of the eigenspace. It is known that the fisherface is superior to the eigenface approach when the training images contain much variation in an individual class; otherwise the performances of fisherface and eigenface are not significantly different [3]. This paper presents a new face representation and recognition method that employs the texture representation derived from the Trace transform. A new distance measure is also proposed by incorporating the weighted Trace transform in order to select only the significant features from the Trace transform.

The organization of this paper is as follows. Section 2 introduces a method for tracing line on an image and some trace functionals we used in this paper. A new texture representation of face images is presented in section 3 . In section 4 we describe a method for weighting the tracing line in the masked Trace transform, to produce the weighted Trace transform. Section 5 presents our experimental results. Finally, we conclude in section 6.

2. The Trace Transform

The Trace transform [1], a generalization of the Radon transform, is a new tool for image processing which can be used for recognizing objects under transformations, e.g. rotation, translation and scaling. To produce the Trace transform one computes a functional along tracing lines of an image. Each line is characterized by two parameters, namely its distance p from the centre of the axes and the orientation ϕ the normal to the line has with respect to the reference direction. In addition, we define parameter t along the line with its origin at the foot of the normal. The definitions of these three parameters are shown in figure 1. The image is transformed to another image with the Trace transform which is a 2-D function depending on parameters (ϕ, p). Different Trace transforms can be produced from an image using different trace functionals. An example of the Trace transform is shown in figure 2. It is shown that the image space in the x and y directions is transformed to the Trace transform space in the ϕ and p directions.

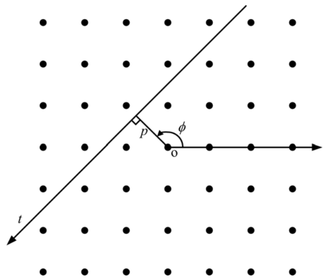

Fig. 1. Tracing line on an image with parameters ϕ, p, and t.

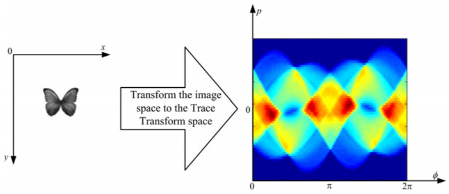

Fig. 2. An image and its Trace transform.

One of the key properties of the Trace transform is that it can be used to construct features invariant to rotation, translation and scaling. We should point out that invariance to rotation and scaling is harder to achieve than invariance to translation. Let us assume that an object is subjected to linear distortions, i.e. rotation, translation and scaling. It is equivalent to saying that the image remains the same but viewed from a linearly distorted coordinate system. Consider scanning an image with lines in all directions. Let us denoted the set of all these lines with Λ. The Trace transform is a function g defined on Λ with the help of T which is some functional of the image function when it is considered as a function of variable t. T is called the *trace functional*.

$$g(\phi, p) = T(F(\phi, p, t)), \tag{1}$$

where $F(\phi, p, t)$ stands for the values of the image function along the chosen line. Parameter t is eliminated after taking the trace functional. The result is therefore a 2-D function of parameters ϕ and p and can be interpreted as another *image* defined on Λ. The resultant Trace transform depends on the functional we used. Let us denote $t_i \in t$ the sampling points along a tracing line defined by ϕ and p. Let us also denote by n the number of points along the tracing line. n may be varied depending on the length of the tracing line. The trace functionals used in our experiments are shown in table 1. The denomination median$_x\{x, w\}$ means the weighted median of sequence x with weights in the sequence w. For example, median$\{\{4, 2, 6, 1\}, \{2, 1, 3, 1\}\}$ indicates the median of numbers $4, 2, 6$, and 1 with corresponding weights $2, 1, 3$, and 1. This means the standard median of the numbers $4, 4, 2, 6, 6, 6, 1$, i.e. the median of the ranked sequence $1, 2, 4, 4, 6, 6, 6$ is 4. See [1] for more details and the properties of the Trace transform.

3. A New Face Representation

The Trace transform is a global transform, applicable to full images. If we are going to use it to recognize faces, we must consider a local version of it. Having identified the faces in an image, we proceed here to define the masked Trace transform. Let us denote by P_i the set of points that make up the boundary of an elliptic face mask [6]. The masked face can be extracted by applying a *filling algorithm* on the set of P_i points. We then obtain the masked face and use it to compute the masked Trace transform. Figure 3 shows the masked face in an oval shape.

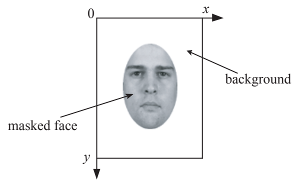

Fig. 3. The masked face in an oval shape.

The Trace transform is known to be able to pick up shape as well as texture characteristics of the object it is used to describe. As we choose all faces we wish to recognize to have elliptical shape, we must be careful to eliminate any shape information from the computation of the Trace transform. For this purpose we work as follows: After we have created an elliptical mask for the face in question, we define a local coordinate system with its centre at the centre of the ellipse. We also define the minimum enclosing rectangle of the ellipse and we flag all pixels that belong to the interior of the ellipse. The tracing lines are defined in the minimum enclosing rectangle as consisting of only the flagged points of the ellipse. The points outside the ellipse are as

Table 1. Examples of the Trace functionals T

No.	Trace Functionals	Details
1	$T(f(t)) = \int_0^\infty f(t)dt$	
2	$T(f(t)) = \left[\int_0^\infty \lvert f(t)\rvert^p \, dt\right]^q$	$p = 0.5, q = 1/p$
3	$T(f(t)) = \text{median}_t\{f(t), \lvert f(t)\rvert\}$	
4	$T(f(t)) = \int_{n+1}^{2n} \left\lvert \frac{d}{dt}\mathcal{M}(\mathbf{F}(t))\right\rvert dt$	$\mathcal{M}$ is a median filtering operator, $\mathbf{F}(t) = [f(t)\ f(t)\ f(t)]$ so that the length of vector $\mathbf{F}(t)$ is $3n$
5	$T(f(t)) = \text{median}_{t^*}\{f(t^*), \lvert f(t^*)\rvert^{1/2}\}$	$f(t^*) = [f(t_{c*})\ f(t_{c*+1})\ \ldots\ f(t_n)]$, $l = 1, 2, \ldots, n$, $c = \text{median}_l\{l, f(t)\}$, $c*$ signifies the nearest integers of c
6	$T(f(t)) = \left\lvert \int_{c*+1}^\infty e^{ik\log(r)} r^p f(t)dt \right\rvert$	$p = 0.5$, $k = 4$, $r = \lvert l - c\rvert$, $l = 1, 2, \ldots, n$, $c = \text{median}_l\{l, \lvert f(t)\rvert^{1/2}\}$, $c*$ signifies the nearest integers of c

if they do not exist for the Trace transform. The shifting of the axes performed ensures translation invariant calculations and discarding of non-flagged pixels (as opposed to simply assuming that they have value 0) ensures that the exact shape of the ellipse plays no role and that only genuine face information is used in the subsequent calculations.

We show the result of the Trace transform of the full image, the rectangular face and elliptical shape in Fig. 4. One can see that the inclusion of background sections destroys the local texture and structure of the Trace transform. However, with the masked Trace transform we can keep the local structure of the shape as well as texture characteristics. We therefore call this face representation the masked Trace transform (MTT). The values of the masked Trace transform may be regarded as some sort of “texture” characteristics of a face image.

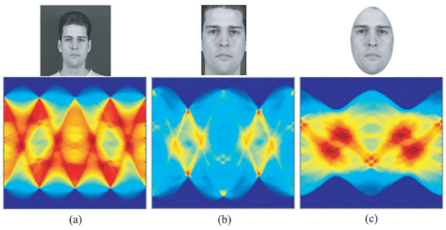

Fig. 4. Examples of the Trace transform. (a) full image (b) rectangular shape and (c) elliptical shape.

In face recognition, one may have the problem of recognizing a face that is rotated, scaled and translated with respect to the face in the database. We show an original face and its rotated versions in figure 5. It can be seen that the masked Trace transform is shifted left or right by the corresponding amount of rotation. One can apply a matching algorithm on each column between an original and rotated versions of the masked Trace transform, i.e. along the ϕ direction, then shift the columns according to the best matching position to the masked Trace transform of the original face. The masked Trace transform of the rotated face can then be reconstructed to match the masked Trace transform of the original face. The procedure is indicated in figure 6. It is therefore seen that the rotation in image space is equivalent to the translation in the Trace transform space. Thus, it is easy to use a matching algorithm between the masked Trace transform of an original face and that of its distorted version to infer the rotation angle.

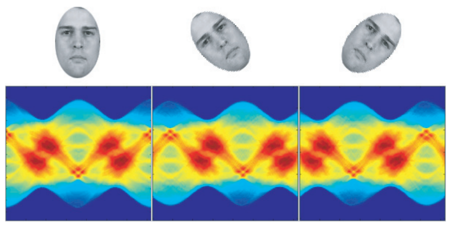

Fig. 5. An original face and its versions distorted by rotation. The corresponding masked Trace transform for each face version is shown in the bottom row.

4 The weighted Trace Transform

The masked Trace transform allows us the define a new way for face coding. Figure 7 shows the masked Trace transforms for 2 different persons in the database and for 4 different images of each person. We observe that there are subtle differences between the Trace transforms of different people that make them distinct, while the Trace transform of the same person seems to retain its structure over the three different images.

Every point in the Trace representation of an image represents a tracing line. Here, we shall describe a method for

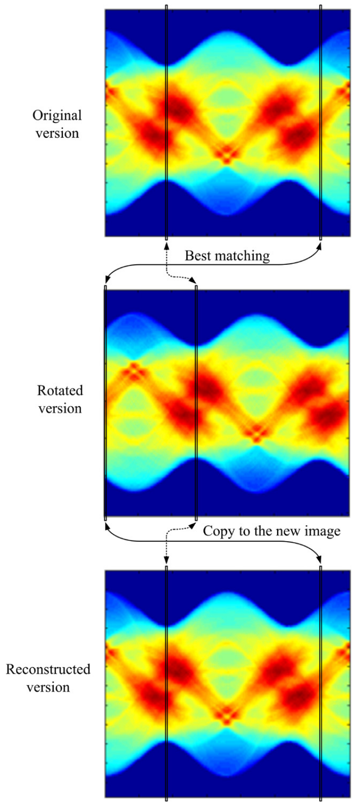

Fig. 6. The reconstruction of the Trace transform.

weighting each tracing line according to the role it plays in recognizing the face. We need to find the persistence of the features locally in the masked Trace transform for each person. So, by selecting the features in the Trace transform which persist for an individual, even when their expression changes, we are identifying those scanning lines that are most important in the definition of the person. We refer this method as the weighted Trace transform. Let us define parameters for weighted Trace transform as follows: F_{ij} is training image j of person i, g_{ij}^k the masked Trace transform for training image j of person i using trace functional k, D_l^{ik} the lth absolute difference of the masked Trace transforms for person i and functional k. $W^{ik}(\phi,p)$ the weighted Trace transform for person i and functional k. The weighted Trace transform can be described step by step as follows.

Step 1. Compute the masked Trace transform, g_{ij}^k, for person i, training image j and functional k.

$$g_{ij}^k(\phi,p) = T_k(F_{ij}(\phi,p,t)), \tag{2}$$

where T_k is the kth Trace functional.

Step 2. Calculate the differences of the masked Trace transforms for person i and functional k.

$$\begin{aligned} D_1^{ik} &= \left|g_{i1}^k - g_{i2}^k\right|, \\ D_2^{ik} &= \left|g_{i1}^k - g_{i3}^k\right|, \\ D_3^{ik} &= \left|g_{i2}^k - g_{i3}^k\right|. \end{aligned} \tag{3}$$

step 3. Compute the weight matrix for the Trace transform for person i, and functional k.

$$W^{ik}(\phi,p) = \begin{cases} 1, & \text{if } \sum_l \rho(D_l^{ik}(\phi,p)) = 0, \\ 0, & \text{Otherwise.} \end{cases} \tag{4}$$

where

$$\rho(x) = \begin{cases} 0, & \text{if } x \le \tau, \\ 1, & \text{Otherwise,} \end{cases} \tag{5}$$

and τ is some threshold. In other words, the weight matrix flags only those scanning lines which in all three images produced values for the Trace transform that differ from each other only up to a certain tolerance τ. When constructing features from the Trace transform we shall use only the flagged lines.

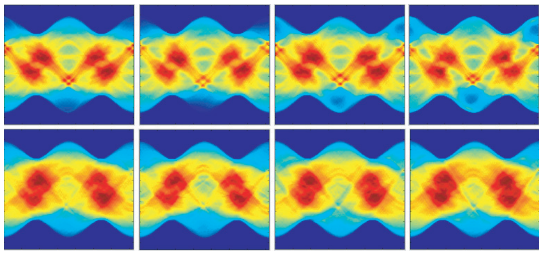

Fig. 7. An example result of the masked Trace transform for different persons. The masked Trace transforms for each person are shown in each row. From left to right, the masked Trace transform for training image 1 to 4 shown in each column.

4.1 Distance measure

Let us explain how test claims are performed for weighted Trace transform. Let us denote by G_r and G_t the reference and test images of the masked Trace transform. We compute the distance between the two images as follows.

$$\mathcal{D} = \max_i\{D(G_r^i, G_t)\}, \tag{6}$$

where $D(G_r^i, G_t) =$

$$\frac{1}{\exp\left[\frac{1}{n_\kappa}\sum_{\phi,p} W(\phi,p)\cdot \min_{\phi_t}\left|g_r^i(\phi,p) - g_t(\phi_t,p)\right|\right]}, \tag{7}$$

where G_r^i is the reference masked Trace transform in training image i, and n_κ the total number of flagged line in weighted Trace transform. We measure the minimum distance between $g_r^i(\phi, p)$ and $g_t(\phi_t, p)$ by which the Trace transform of the test image is scan along the ϕ_t direction. This is the same procedure as indicated in figure 6. The weight $W(\phi, p)$ is computed using (4). $D(G_r^i, G_t)$ returns the highest value 1 when G_r^i and G_t is exactly the same.

5. Experimental Results

In this section, we describe a face database we used and then present a face recognition result under rotation, size variation and facial expression. Our proposed method was implemented on the AR face database [2]. The database contains color images corresponding to 126 people (70 men and 56 women). The pictures were taken at the CVC under strictly controlled conditions. No restrictions on wear (clothes, glasses, etc.), make-up, hair style, etc., were imposed to participants. Each person participated in two sessions, separated by two weeks time. The same pictures were taken in both sessions. Some examples of face images used in training procedure are shown in figure 8.

Fig. 8. Some examples of face images in AR database.

In our current implementation, we have 6 different trace functionals. Some examples of the Trace transform computed from 6 different functionals are shown in figure 9. It should be pointed out that the results from all trace functionals may be thought of as texture representations which can be used as a robust feature for recognition task. We however use a trace functional for a specific person. On the other hand, the choice of the trace functionals we used for a particular person depends on the classification ability in which the between-class variance is maximized with respect to the minimized within-class variance. In addition, by using the technique we proposed, WTT, the redundant features are ignored in the computational process. This helps us maximize the matching confidence between individual face images, even when their expression changes.

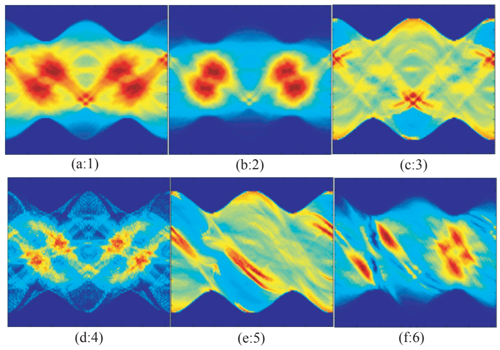

Fig. 9. An example of the masked Trace transforms computed using the 6 different trace functionals of table 1.

5.1. Face Recognition under rotation, size variation and facial expression

In the real world applications, the face recognition should be invariant to rotation, size variation and facial expression. In our proposed method, we use 3 images from the face database for training. The face images for testing were generated by applying a random scaling and rotation factors to the face images, which was distributed within $[1 - 50, 1 + 50]\%$ and $[0, 360]$ degree. Examples of test images are shown in figure 10. The rightmost image is an extreme case of rotated image which is hard to recognize even for the human ability.

Fig. 10. Examples of cropped face images under rotation, size variation and facial expression.

In general, the face registration method was used in the preprocessing step to locate the faces in an image. The located faces were then normalized (in scale and orientation) for which the two eyes were aligned at the same position. This is hard to achieve in the real time application in which the corrected eye position can not be found. In contrast, the face registration is not used here. In our method the rotated and scaled versions of face images were directly used in the recognition task. We then use the distance measure described in section 4.1 to recognize the face images. Figure 11 shows results of the masked Trace transform of face images under rotation, size variation and facial expression.

It can be seen that the 'images' in the Trace transform space remain the same but it is viewed as the shifted version of the original image for the rotation problem. The texture representation is also robust to scaling condition of face image. In fact, the 'images' in the Trace transform space are affected by some scaling factor. The more values of scaling factor, the more decreases of recognition performance. Table 2 shows results of face recognition with top 5 best matching images. In the top 5 classification, the corrected match is counted when the individual face of input image is among the best 5 matched faces. In table 2, we also show the results of facial expression in 3 cases: smile, anger and scream. Smiling and angry expression decrease the recognition rate by some amount. The scream expression is a special case of local deformation of face images in which the geometric location (e.g. mouth position) and texture information are destroyed (second rightmost image in figure 10). In this case we obtain a recognition rate of 35.56%. In summary, our proposed method is robust to rotation, size variation and facial expression. From the inspection of the table 2, it was found that our proposed method performed better than the eigenface method in all cases. Such robustness comes from the use of masked Trace transform, weighted Trace transform and matching measure in (7). It also comes from the fact that only the flagged line is used rather than entire face representation which helps us maximize the matching between reference and test images. Another advantage of our approach is that, when new subjects are added to the system we do not need to retrain on the whole face database, in fact only images of the new subject are used to find the optimal parameter τ of the algorithm. This may not be the case for eigenface: when new subjects are added to the face database, these systems must be retrained over the whole face database, which is a barrier for real applications.

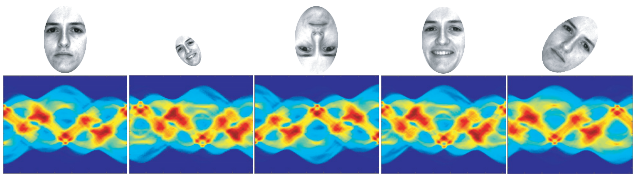

Fig. 11. The face images under rotation, size variation and facial expression. The corresponding face representations are shown in bottom row

6. Summary and Conclusions

We have presented a new face representation using texture characteristic derived from the Trace transform. Texture representation, masked Trace transform, provides classification ability for reducing the within-class variance of an individual face with the help of weighted Trace transform.

Table 2. Performance Comparison of Face Recognition under Rotation, Scaling and Facial Expression.

Testing Faces	Method	
	EigenFace	MTT+WTT
Scaling	68.8	77.3
Rotation	55.4	96.2
Scaling+Rotation	46.8	71.1
Smiling Expression	82.04	86.3
Angry Expression	73.21	91.53
Screaming Expression	32.14	35.56

The masked Trace transform captures the key characteristics of face images by suppressing variations of a face, while maintaining discriminability. In addition, the redundant features are ignored by using only the flagged line in the masked Trace transform. We obtain a high recognition rate which is outperformed the eigenface method in all cases.

Reference

[1] A. Kadyrov and M. Petrou, "The Trace Transform and Its Applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 8, pp. 811-828, August 2001.

[2] A. M. Martinez and R. Benavente, "The AR Face Database," CVC Technical Report no. 24, 1998.

[3] C. Liu and H. Wechsler, "A Shape- and Texture-Based Enhanced Fisher Classifier for Face Recognition," IEEE Transactions on Image Processing, Vol. 10, No. 4, pp. 598-608, Apr. 2001.

[4] M. Turk and A. Pentland, "Eigenfaces for Recognition," Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71-86, 1991.

[5] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp. 711-720, Jul. 1997.

[6] S. Srisuk and W. Kurutach, "A New Robust Face Detection in Color Images," *in Proc. IEEE Int. Conf. on Automatic Face and Gesture Recognition*, Washington D.C., USA, pp. 306-311, May 20-21, 2002.