Robust Hausdorff distance measure for face recognition

Vivek E. Pa, N. Sudha

Synopsys India, Bangalore 560 016, India
School of Computer Engineering, Nanyang Technological University, Singapore-639798, Singapore

Received 1 July 2005; received in revised form 12 January 2006; accepted 19 April 2006

Abstract

Face is considered to be one of the biometrics in automatic person identification. The non-intrusive nature of face recognition makes it an attractive choice. For face recognition system to be practical, it should be robust to variations in illumination, pose and expression as humans recognize faces irrespective of all these variations. In this paper, an attempt to address these issues is made using a new Hausdorff distance-based measure. The proposed measure represent the gray values of pixels in face images as vectors giving the neighborhood intensity distribution of the pixels. The transformation is expected to be less sensitive to illumination variations besides preserving the appearance of face embedded in the original gray image. While the existing Hausdorff distance-based measures are defined between the binary edge images of faces which contains primarily structural information, the proposed measure gives the dissimilarity between the appearance of faces. An efficient method to compute the proposed measure is presented. The performance of the method on benchmark face databases shows that it is robust to considerable variations in pose, expression and illumination. Comparison with some of the existing Hausdorff distance-based methods shows that the proposed method performs better in many cases.

Keywords: Hausdorff distance; Face recognition; Face image

1. Introduction

Hausdorff distance was originally defined as a dissimilarity measure on point sets. It later got wide acceptance in image comparison [1,2]. A more specific application is proposed by Huttenlocher et al. [3] for object detection and recognition. Their work contains a modification to the actual Hausdorff distance called partial Hausdorff distance (PHD). This modification is meant to get distance measure between the most closely matching portions of the images being compared which in turn reduces the effect of occlusion in object matching. In object recognition, as shape information plays a decisive role, Hausdorff distance between edge images is a suitable measure for comparison.

Many variants of Hausdorff distance have been proposed for face recognition. One of them, namely modified Hausdorff distance [4] uses a neighborhood function and penalty value to give preference to points which are located within a given neighborhood. Guo et al. [5] proposed spatially weighted Hausdorff distance (SWHD) as an improvement to conventional Hausdorff distance between edge images. As the name suggests, this method gives different weights to facial regions while finding the Hausdorff distance between edge images. This is done under the assumption that the facial regions like eyes, mouth, etc. have more importance in recognizing faces compared to other regions. A weight function with rectangular regions of different sizes and weights which can be overlapped with the face images is used for this purpose. Eigenface was suggested in Ref. [6] as a better choice for weight function than the fixed rectangular window function. In recognizing faces, the portions of face
images that varies prominently for different subjects have more importance. The components of eigenfaces obtained from a set of reference faces of subjects will have large values in the regions where the images have prominent changes.

Gao and Leung [7] proposed a different approach for face recognition. In their work, line segments extracted from face edge curves are used as features. For comparing faces, they have defined three distances between line segments. Later Gao proposed a distance measure based on features called 'dominant points' [8]. Dominant points are the points extracted from the portions of edge maps having high curvature. While finding the dominant points, a merit factor is associated with each of them. These factors are incorporated while determining the Hausdorff distance. As faces are represented by a small set of dominant points, the storage of faces takes less space.

All the above mentioned methods for face recognition use edge images for finding Hausdorff distance or its variants. Some of them are computationally intensive as they extract additional features like eigenfaces and dominant points. As edge images contain shape information, they are suitable features for object detection and recognition. However, when face recognition is concerned, the appearance is important than the edge maps. It has been established in Ref. [9] that when humans recognize faces, the features like cheeks, jaw, etc. have equal importance as features like mouth, eyes, etc. The overall appearance of the face is determined by all these features and human recognizes a face based on overall appearance. The intensity distribution of pixels captures this appearance information. However, direct comparison of gray images is not advisable as the performance will be affected by illumination variations unlike edge maps.

In this paper, we propose a new PHD-based measure to compare the appearance of faces. The measure is applied to face recognition based on gray images of faces instead of edge maps. The measure is robust to variations in a face due to expression, illumination and slight pose differences. The partialness in the measure could tolerate expression and illumination changes. It first describes the image transformation and then defines \( H_{p} \) and lists the properties of the measure. A time-space efficient algorithm is proposed for the computation of \( H_{p} \) in Section 4. Its theoretical complexity is also analyzed. Section 5 studies the values of \( H_{p} \) for face recognition. In Section 6, the performance of \( H_{p} \) for face recognition is evaluated using benchmark face databases. Comparison with existing methods is given in Sections 7 and 8 concludes the paper.

2. Conventional Hausdorff distance

The Hausdorff distance is defined as a distance between two point sets. This distance gives a measure of dissimilarity between the point sets. Let \( A = \{a_1, a_2, \ldots, a_m\} \) and \( B = \{b_1, b_2, \ldots, b_n\} \) be two point sets. Then the Hausdorff distance between \( A \) and \( B \) is given by

\[
H(A, B) = \max(h(A, B), h(B, A)),
\]

where \( h \) is the directed Hausdorff distance which is given by

\[
h(A, B) = \max \min_{a \in A, b \in B} \| a - b \|.
\]

\( \| \cdot \| \) is the norm of a vector and \( h(A, B) \) and \( h(B, A) \) are different. The directed Hausdorff distance \( h(A, B) \) finds the point \( a \in A \) whose distance from its nearest point in \( B \) is maximum among all points in \( A \) and gives the distance between \( a \) and its nearest point in \( B \). When \( h(A, B) = d \) all points in \( A \) will have at least one point in \( B \) within a distance of \( d \) and there will be at least one point in \( A \) whose nearest point in \( B \) is exactly at a distance of \( d \). Thus as a dissimilarity measure from \( A \) to \( B \), \( h(A, B) \) gives the distance between the most mismatching point in \( A \) and its nearest point in \( B \). Hausdorff distance \( H(A, B) \) takes the maximum of the directed distances from \( A \) to \( B \) and from \( B \) to \( A \). When the Hausdorff distance is taken between images, point sets are replaced with pixel sets containing the coordinates of feature pixels.

One advantage of using Hausdorff distance for shape comparison is that it does not require the explicit pairing of points in \( A \) and \( B \). When Hausdorff distance is used directly for comparing objects, the mismatch will be large when a part of an object is missing due to occlusion or when there are outliers. This is undesirable in object matching. To overcome this, a modification of Hausdorff distance to compare between the closely matching portions of the objects is available [3]. This modified measure is called PHD. It takes the \( K^{th} \) ranked maximum value instead of the overall maximum in the directed Hausdorff distance.

\[
H_{p}(A, B) = \max(h_{p}(A, B), h_{p}(B, A)),
\]

where \( h_{p}(A, B) \) is the partial directed Hausdorff distance and is given by

\[
h_{p}(A, B) = K^{th} \min_{a \in A, b \in B} \| a - b \|.
\]

3. New partial Hausdorff distance-based measure \( (H_{pv}) \) for comparing faces

Existing Hausdorff distance-based measures for face recognition are defined between edge images of faces.
Edge images are less affected by illumination variations. However, edge images do not carry the overall facial appearance. When the gray images that have appearance information are directly considered for comparison, its performance is affected by illumination variations. The effect of illumination can be reduced to a large extent by representing a pixel based on relative intensities of pixels in its neighborhood. Thus, a pixel is represented by a vector rather than a single gray value. This section defines a PHD-based measure to compare the transformed face images. We put forth first the notations used in this section.

3.1. Notations

- \( M \) and \( T \) Model and test images
- \( M_t \) and \( T_v \) Transformed images corresponding to \( M \) and \( T \)
- \( g(p) \) Gray value of pixel \( p \)
- \( v(p) \) Vector representation of pixel \( p \)
- \( d(\cdot , \cdot) \) Distance measure between two quantities
- \( H_e \) New Hausdorff distance-based measure
- \( H_{pv} \) New PHD-based measure
- \( h_v \) Directed version of \( H_e \)
- \( h_{pv} \) Directed version of \( H_{pv} \)
- \( f \) Partialness fraction
- \( K \) Rank corresponding to \( f \)
- \( r \) and \( c \) Number of rows and columns of an image

3.2. Image transformation

A representation of gray images which is tolerable to illumination variations besides preserving the appearance information is necessary for the comparison of faces images robust to illumination changes. The robustness to expression and slight pose variations can be achieved by incorporating partialness in the measure. In a gray image, the sign of first-order derivative operation at each pixel is expected to remain same for a wide range of illumination variations. The first-order derivative at a pixel is approximated to the difference between the intensity values of the pixel and its neighbor along a direction. The representation of a pixel in terms of sign of first derivative, in some arbitrary direction, at that pixel point is not much useful in comparing faces in images. A more useful representation can be obtained by considering the 8-neighborhood of the pixel. Consider a pixel and its 8-neighborhood which forms a \( 3 \times 3 \) window. The signs of the first derivative taken along the direction of neighbors is expected to remain same for uniform illumination changes over the window. The advantage of representing a pixel in terms of its neighborhood is that it captures the distribution of the intensities in the neighborhood. This information acts as a signature for the pixel in finding the matching pixels when face images are compared.

In order to achieve a performance independent of illumination and lighting conditions, the gray image is transformed into an image whose pixels are assigned an 8-element vector.

Each element of the vector corresponds to the first derivative along the direction of a neighbor of the pixel. The element takes one of the values, viz., 1, 0 and −1. Let \( g(p) \) and \( g(p_n) \) be the gray values of pixels \( p \) and its neighbor \( p_n \). Then the element (corresponding to \( p_n \)) of vector assigned to \( p \) takes the value 1 if \( g(p) > g(p_n) \); 0 if \( g(p) = g(p_n) \); −1 if \( g(p) < g(p_n) \). Fig. 1 shows a \( 3 \times 3 \) window in an image and the vector corresponding to the pixel \( p \) at the center. The transformed image preserves the intensity distribution and hence the appearance of face besides insensitive to illumination.

For the purpose of plotting the transformed image, each vector is uniquely represented by a color in RGB space and a color image is obtained. A vector in the transformed image have eight components each having value −1, 0 or 1. The green component of the color assigned to the vector is obtained by setting the components of a vector with value −1 to 1 and all other vector components to 0. This results in an 8-bit binary number. For example, the green component’s value of the vector \([-1, 0, 0, 1, 1, -1, -1, 0] \) is given by \((10000110)_2 \) or \((134)_{10} \). Similarly, red component is obtained by setting the components of vector with value 0–1 and the remaining components to 0, and the blue component is obtained by setting the components of vector with value 1–1 and the remaining components to 0. The green and blue component values of the example vector taken are \((01100001)_2 \) and \((000110000)_2 \), respectively.

The robustness of the proposed transformation to illumination variation is illustrated using images from CMU PIE face database. For comparing the perceptual appearance of the color image, a metric called S-CIELAB \([10,11]\) which is a spatial extension of CIELAB color metric is used. It gives the error between two color images based on the difference in the perceptual appearance. The pixel intensities give the amount of perceptual error. The white pixel corresponds to maximum error and the black corresponds to minimum error. Fig. 2(a) and (b) show two images of a person taken under two different illumination conditions and Fig. 2(c) shows the error image computed between them. The color images are then converted into gray images (Figure 3(a) and (b)) and their transformed images are shown in Fig. 3(c) and (d). Fig. 3(e) is the error image between these two transformed images. The error image is almost dark which shows that the transformed images are less affected by illumination.

\[
\begin{array}{cccc}
74 & 74 & 76 & 1 \\
94 & 96 & 96 & 1 \\
111 & 109 & 105 & -1 \\
\end{array}
\]

Fig. 1. (a) Gray values of pixels in a neighborhood; (b) elements of vector corresponding to the neighbors; (c) vector representation of \( p \).
variations. The effectiveness of the transformation for comparing faces in different illumination is hence verified by computing S-CIELAB error image. Fig. 4(a) and (b) show images of two different persons captured under normal illumination. Fig. 4(c) and (d) are the corresponding transformed images. The error image between 4(c) and (d) is shown in Fig. 4(e). It has larger values over nose, mouth and eye regions and hence discriminate the transformed images of different faces. It is hence observed that the transformed images are suitable for face recognition. The new PHD-based measure $H_{pv}$ is defined between the transformed images of faces.

### 3.3. Definition of $H_{pv}$

Let $M$ and $T$ be the face images of size $r \times c$ to be compared. Let $M_v$ and $T_v$ be the transformed images. The border pixels of $M$ and $T$ are ignored for the transformed images as they do not have all eight neighbors. The size of $M_v$ or $T_v$ is therefore $(r-2) \times (c-2)$. Let $v(p)$ represent the vector at pixel $p$. Then the new Hausdorff distance-based measure between $M$ and $T$ is defined as

$$H_v(M, T) = \max(h_v(M, T), h_v(T, M)), \quad (5)$$

where $h_v$ is the directed version of $H_v$ and is given by

$$h_v(M, T) = \max_{m \in M_v} \min_{t \in T_v} d(m, t), \quad (6)$$

where

$$d(m, t) = \begin{cases} \|m - t\| & \text{if } v(m) = v(t), \\ L & \text{if } v(m) \neq v(t), \end{cases} \quad (7)$$

$v(m) = v(t)$ when all the components of $v(m)$ and $v(t)$ are equal. $L$ is a large value which can be $\sqrt{r^2 + c^2} + 1$. $l_2$ norm has been chosen for finding distance between $m$ and $t$. The computation of $h_v(M, T)$ leads to the assignment of a distance value to every pixel $m \in M_v$ and $h_v(M, T)$ is the maximum of these distance values. The effect of directional lighting, shadowing, occlusion and local expression variations results in poor performance of proposed measure in face recognition.

The proposed measure can be improved for image comparison applications by introducing partial matching as given in Ref. [3]. Partial matching is obtained by taking $K^{th}$ ranked distance instead of maximum in $h_v$. The equations are given by

$$h_{pv}(M, T) = K^{th} \min_{m \in M_v, t \in T_v} d(m, t), \quad (8)$$

$$H_{pv}(M, T) = \max(h_{pv}(M, T), h_{pv}(T, M)), \quad (9)$$

where $K = f \times \text{size}(M)$ for $h_{pv}(M, T)$ and $K = f \times \text{size}(T)$ for $h_{pv}(T, M)$. $f$ is a given fraction. $H_{pv}$ has the following properties.
3.4. Properties of $H_{pv}$ and related quantities

1. If $h_v(M, T) = \hat{d}$, then every pixel $p$ in $M_v$ is at a distance less than or equal to $\hat{d}$ from some pixel $p'$ in $T_v$ with $v(p') = v(p)$.

2. If $h_{pv}(M, T) = \hat{d}$ with a partialness of fraction $f$, then $h_v$ between every subset of pixels of $M_v$ of size $f \mid M \mid$ and the entire $T_v$ is greater than or equal to $\hat{d}$. $h_{pv}$ corresponds to the subset whose $h_v$ is minimum. In other words, $h_{pv}$ automatically selects the best matching portion of $M_v$ (of size $f \mid M_v \mid$ pixels) that minimizes the distance value of the directed version of new measure with $T_v$.

3. $h_v$, $H_v$, $h_{pv}$ and $H_{pv}$ are all positive. $H_v = 0$ if the transformed images $M_v$ and $T_v$ are same.

4. $h_{pv}(M, T) = 0 \iff h_{pv}(T, M) = 0$.

5. $h_{pv}(\ldots, \ldots) = 0 \iff H_{pv}(M, T) = 0$.

6. If $h_{pv}(\ldots, \ldots) = 0$, then there is a common best matching portion in both $M_v$ and $T_v$ with same vector values.

7. If one vector in $M_v$ is different from all vectors in $T_v$, then $h_v(M_v, T_v)$ as well as $H_v(M_v, T_v)$ take the value $L$. The different vector is mainly due to the effect of background and noise. $H_{pv}(M, T)$ can tolerate these effects. $h_{pv}(M, T) = L$ only if more than $(1 - f)r c$ vectors in $M_v$ are different from vectors in $T_v$.

8. Let $SM_v$ and $ST_v$ represent the sets of vectors in $M_v$ and $T_v$, respectively. If $SM_v$ and $ST_v$ are equal, then $H_v \neq L$. If $|SM_v - (SM_v \cap ST_v)|$ or $|ST_v - (SM_v \cap ST_v)|$ is greater than $(1 - f) r c$, then $H_{pv} = L$.

4. Computation of $H_{pv}$

A time and space efficient algorithm is proposed to compute $H_{pv}$. The computation of $h_{pv}(M, T)$ in Eq. (8) involves finding the nearest pixel $t$ in $T_v$ for each pixel $m$ in $M_v$ with $v(m) = v(t)$. For $n \times n$ images, the direct computation takes $O(n^4)$ time as each pixel $m$ in $M_v$ requires one scan of image $T_v$. Instead of scanning the entire image, if linked list of pixels in $T_v$ is constructed for every possible vector, then the search for the nearest pixel can be limited. There are $3^8$ possible vectors and hence an array of $3^8$ linked lists are constructed. The elements of array are the pointers to the linked lists. The array index $i$ is computed from a vector $[a_0, \ldots, a_7]$ as $i = \sum_{k=0}^{7}(a_k + 1) \times 3^k$. The pictorial representation of the data structure is shown in Fig. 5.

Once the data structure is created for $T_v$, the computation of $h_{pv}(M, T)$ is done as follows. For each pixel in $M_v$, the linked list corresponding to its vector value is searched linearly for the nearest pixel and the distance to it is assigned. This leads to the assignment of distance value to every pixel in $M_v$. The $K^{th}$ ranked distance is then computed. This gives $h_{pv}(M, T)$. Similarly $h_{pv}(T, M)$ is computed by creating data structure for $M_v$ and $H_{pv}(T, M)$ is the maximum of these two distances.

4.1. Algorithm

**Inputs:** Face images $M$ and $T$ of size $r \times c$

**Output:** $H_{pv}$

$M_v = VectorImage(M)$

$T_v = VectorImage(T)$

{Construct data structures as in Fig.5 for $M_v$ and $T_v$}

for $j = 0 : (r - 3)$

for $l = 0 : (c - 3)$

$i = index(M_v[j][l])$

Add($P[i,j], j, l$)

$i = index(T_v[j][l])$

Add($Q[i], j, l$)

end for

end for

{Compute distance values for pixels in $M_v$ and $T_v$}

$s = 0$

for $j = 0 : r - 3$

for $l = 0 : c - 3$

$i = index(M_v[j][l])$

$dist = FindNearest(Q[index], j, l)$

$F(s) = dist$

$i = index(T_v[j][l])$

$dist = FindNearest(P[index], j, l)$

$R(s) = dist$

$s = s + 1$

end for

end for

$d = rank(F, K)$

$e = rank(R, K)$

$H_{pv} = \max(d, e)$

In the algorithm to compute $H_{pv}$, the function $VectorImage()$ converts a gray level image to its vector representation. The function $index()$ computes index value of a vector for accessing a linked list in the data structure. $Add()$ will insert a pixel to the beginning of a particular linked list. $P$ and $Q$ represent the data structure shown in Fig. 5 corresponding to $M_v$ and $T_v$. $FindNearest()$ finds the nearest pixel to a given pixel from a list of pixels. To find $h_{pv}(M, T)$, for
each pixel in $M_v$ distance to the nearest matching pixel in $T_v$ is determined and stored in $F$. Similarly the distance values for $h_{pv}(T, M)$ is stored in $R$. The maximum of $K^{th}$ ranked distances $d$ and $e$ of $F$ and $R$ arrays is the desired distance measure $H_{pv}$.

4.2. Complexity analysis

4.2.1. Time complexity

The vectorization of the image of size $r \times c$ can be done in a single scan and hence the time complexity of $\text{VectorImage}()$ is $O(rc)$. $\text{index}()$ can be computed in constant time. Insertion of all pixels in the images to corresponding lists takes $O(rc)$ time. As $\text{FindNearest}()$ involves linear search of a list of pixels, the time taken by this operation depends on the length of the list. To find $H_{pv}$ between two images, $\text{FindNearest}()$ has to be executed $2rc$ times. Hence, the time required to find distance will be $O(drc)$ where $d$ is the length of the largest list. There exists algorithms [12] to find $K^{th}$ ranked value from an unsorted array in linear time and hence $\text{rank}()$ has a time complexity of $O(rc)$. The entire algorithm’s time complexity is therefore $O(drc)$. The value of $d$ can be 1 when $rc \leq 3^8$ and will be $rc$ when all the pixels in an image have the same vector value.

The actual computation time has been taken for face images of different sizes. For images of each size, more than thousand computations were done and the average time of all such computations were taken. The algorithm has been implemented in C programming language and the experiments were done on a uniprocessor system (Intel Pentium(R)-4, 1.7GHz, 1GB RAM) running Linux operating system. The average time taken for the computation of $H_{pv}$ for images of different image sizes are displayed as bar chart in Fig. 6.

4.2.2. Space complexity

The space required by the images is $O(rc)$. The same space can be utilized for storing transformed images as original images are not used for further computation. The array of pointers to the lists of pixels is of size $2 \times 38$. This is constant independent of image size. As all the pixels in both the images will be added once to lists of pixels, the total memory used in constructing the data structures for images is $2(3^8 + rc)$ units. The arrays $F$ and $R$ used to hold the distance values occupy a memory of $rc$ units each. Hence the asymptotic space complexity of the algorithm is $O(rc)$. As data structures $P$ and $Q$ need not be computed simultaneously, the actual memory usage can be reduced by using same array for both $P$ and $Q$. Similarly, a large array of length $rc$ is sufficient for both $F$ and $R$.

5. Suitability of $H_{pv}$ for face recognition

Recognition based on the new measure $H_{pv}$ is expected to be robust to expression changes, pose differences and illumination variations in face images. A frontal face with only expression variations can be viewed as local variations in the image. As $H_{pv}$ can perform partial matching, face recognition invariant to expression can be achieved. Pose variation can cause two types of changes in the distribution of vectors over the 2-D plane. (1) It shifts the distribution. (2) It causes some portions of the distribution in frontal face to disappear and creates new portions in the pose-variant
As PHD can tolerate the changes in the positions of the feature points to some extent, it is expected to handle some amount of pose variations. As the new measure is based on the relative intensities of neighbors and not on absolute intensity values, it is invariant to lighting and illumination conditions.

$H_{pv}$ has been computed between several sample faces and their values have been studied for application to face recognition. Ideally, irrespective of variations in a face, the $H_{pv}$ values between same faces and between different faces should be well separated to use it for face recognition. The values of $H_{pv}$ are computed between normal faces and expression as well as illumination varying faces of five different persons. The face images are shown in Fig. 7. The normal faces and their pose varying ones are shown in Fig. 8. Tables 1–3 show $H_{pv}$ values between normal faces and their expression, illumination and pose variations, respectively. In all the tables, the distance values between same faces are found to be smaller than those between different faces. These distance values support the usefulness of $H_{pv}$ for face recognition.

### 6. Performance evaluation of $H_{pv}$ for face recognition

The performance of the new measure was evaluated for face recognition. Face recognition involves classifying a given face image as one of the predefined identities specified by the model face images of subjects. It is done by finding $H_{pv}$ between the given face and each of the model faces and assigning the identity of the model face that gives the smallest $H_{pv}$. The results of the experiments are given as recognition rate which is the ratio of the number of images correctly classified to the total number of images in the test set. Bench mark face databases consisting of face images with these types of variations were used to evaluate the $H_{pv}$-based face recognition system.

#### 6.1. Face databases used for the studies

For comparing the performance of $H_{pv}$ with existing methods, face databases consisting of images with different types of variations are considered. The AR face database [13] from Purdue University consists of images of 126...
subjects (70 men and 56 women). The images were taken in two sessions. The expression and illumination varying images were used for the performance studies. Fig. 9 shows sample images of a subject. The Yale face database [14] consists of 15 subjects each having 11 images. The images have expression variations as well as illumination variations due to a light source positioned at right, left and center with respect to the face. Some of the images of a person are shown in Fig. 10. The Bern face database consists of 30 subjects. Each subject has two images of five different poses. The pose variations are due to face looking straight, downward, upward, left and right. Sample images are shown in Fig. 11.

6.2. Influence of $f$ on the performance of $H_{pv}$

The recognition rate of $H_{pv}$ is expected to be low for both high and low values of $f$. When the value of $f$ is too
Fig. 13. (a) and (d) Normal faces; (b) and (e) Faces with smiling expression; (c) and (f) Faces with angry expression.

low, the $H_{pv}$ is computed between the similar regions in the faces. The features that discriminate the faces are not considered for the computation. This leads to poor recognition rate. When the value of $f$ is too high, the background and variations in expression, pose and illumination affect the performance.

Fig. 12 shows the performance of $H_{pv}$ for various values of $f$ on expression varying face images from AR face database. The performance for screaming expression is found to be lower than other expressions. This is due to the fact that the variations in face due to screaming expression is larger than the variations due to smiling and angry expressions as seen in the samples shown in Fig. 9. The performance of $H_{pv}$ on smiling and angry expressions is almost same for large $f$ and it differs for smaller values of $f$. This has been analyzed. Some correctly classified smiling images and misclassified angry faces are shown in Fig. 13. Though the variation in the smiling face from their normal face is more when compared to the angry face, the angry faces displayed in the figure are misclassified due to the tilt in these faces. It is found that such tilt is less in the case of smiling faces in the database and hence the performance is better for small $f$ that considers small similar portions unaffected by expression.

Fig. 14 shows the recognition rate for different values of $f$ for faces with illumination variations due to directional lighting. The fact that recognition rate starts to fall for moderately high values of $f$ can be explained with the transformed images corresponding to a normal face and its illumination varying images shown in Fig. 15. Due to the effect of point light source focused on to the face, the gray value distribution of a part of a face is disturbed and tend to be uniform. This uniformity in the gray values of facial regions results in red patches in the transformed images shown in the figure. The images which are illuminated from both sides, nearly half of the facial regions are affected. For such images, recognition rate comes down for values of $f$ greater than 0.5. In all the three cases, the recognition rate is found to be quite high for values of $f$ as low as 0.2. The only variation involved in these images is due to directional lighting. Hence classification made by considering even small portions of the images is expected to give good results.

Fig. 14. Recognition results for illumination varying faces in AR face database.
The performance of $H_{pv}$ for different values of $f$ on Yale database is shown in Fig. 16. The Yale database has both expression and illumination varying images. Due to the effect of shadow, in some of the illumination varying images nearly half of the face is darkened. Examples are shown in Fig. 17. This causes fall in recognition rate when the value of $f$ is more than 0.5. For higher values of $f$, the expression variations also affects the performance. For lower values of $f$, recognition rate falls. This is because the closely matching regions of faces are only considered for finding $H_{pv}$ and they do not have the facial parts that discriminate the faces.

The influence of $f$ on $H_{pv}$ was also studied on pose varying faces of Bern face database. When frontal and pose varying face images are compared, there are portions of face that appear in one image and are missing in the other image. When $H_{pv}$ is computed for a large value of $f$, such portions in the images affect the recognition performance. This causes the curve in Fig. 18 to fall for large values of $f$. When $f$ is small, similar to all other cases recognition rate falls. From the plots shown in Figs. 12, 14, 16 and 18, we can observe that good rate is achieved for the values of $f$ around 0.5.

7. Comparison studies with existing methods

The performance of the new measure is compared with PHD on edge maps [3], doubly modified Hausdorff distance (M2HD) on edge maps [4], Hausdorff distance based on line edge map (LEM) [7], SWHD [5] and spatially eigen-weighted Hausdorff distances (SEWHD and SEW2HD) [6]. The results for $H_{pv}$ are produced for $f = 0.5$. This choice of $f$ is based on the overall performance for different
databases. The results for PHD are computed for values of $f = 0.5, 0.55, 0.6, \ldots, 0.9$ and the best recognition rate is reported. Three different databases viz. AR face database [13], Bern face database [15] and Yale face database [14] which were described in the previous section were used for comparison studies.

7.1. Studies with faces varying in expression

The performance of $H_{pv}$ on expression varying face images is compared with some of the existing face recognition methods using AR face database [13]. One normal frontal image corresponding to each person is included in the model set. The test set consists of three images per person with smiling, screaming and angry expressions. The performance of the new method is compared with PHD, LEM and M2HD. The results for M2HD and LEM are obtained from Ref. [7]. The results for different methods are given in Table 4. The performance of $H_{pv}$ is found to be superior to the existing methods. The recognition rate of $H_{pv}$ on faces with screaming expression is found to be less than that of other expressions. This is due to the fact that the variation in face caused by screaming expression is larger than that caused by smiling and angry expressions.

7.2. Studies with faces in different poses

The comparison study on pose varying faces is carried out with Bern face database [15]. For each subject, the frontal face is taken as the model image. Eight pose varying face images are used as test images. The performance of the recognition system based on the new measure is compared with PHD, LEM and M2HD and the results are shown in Table 5. The performance of $H_{pv}$ is found to be better than the existing methods in all cases.

7.3. Studies with faces varying in illumination

The performance of $H_{pv}$ is compared with the existing variants of Hausdorff distance using the face images of Yale
database varying in illumination and expression. The variants are PHD, M2HD, SWHD, SEWHD and SEW2HD. The results are shown in Table 6. The results of the variants of Hausdorff distance are obtained from Ref. [6]. In another study, the illumination varying images from AR face database were used to compare the new method with PHD, LEM and M2HD. There are three illumination varying images per subject, viz., with a light source on left side of face, with a light source on right side of face and with light sources on both sides. The performance of LEM and M2HD are taken from Ref. [7] and the comparison study is reported in Table 7. $H_{pv}$ performs better than the existing methods.

### Table 6
Comparison study with Yale database faces varying in illumination and expression

<table>
<thead>
<tr>
<th>Method</th>
<th>Recognition rate in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHD</td>
<td>84</td>
</tr>
<tr>
<td>M2HD</td>
<td>80</td>
</tr>
<tr>
<td>SWHD</td>
<td>82</td>
</tr>
<tr>
<td>SEWHD</td>
<td>85</td>
</tr>
<tr>
<td>SEW2HD</td>
<td>89</td>
</tr>
<tr>
<td>$H_{pv}$</td>
<td>95.33</td>
</tr>
</tbody>
</table>

### Table 7
Comparison study using illumination-varying images from AR database

<table>
<thead>
<tr>
<th>Test faces with</th>
<th>PHD</th>
<th>LEM</th>
<th>M2HD</th>
<th>$H_{pv}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Left light on</td>
<td>88.88</td>
<td>92.86</td>
<td>82.14</td>
<td>99.2</td>
</tr>
<tr>
<td>Right light on</td>
<td>84.92</td>
<td>91.07</td>
<td>73.21</td>
<td>97.61</td>
</tr>
<tr>
<td>Both lights on</td>
<td>64.28</td>
<td>74.11</td>
<td>54.46</td>
<td>84.12</td>
</tr>
</tbody>
</table>

### 8. Conclusion

A new partial Hausdorff distance-based method for face recognition has been proposed in this paper. The recognition of faces is done by comparing the appearance of faces embedded in the gray images of them. This involves a transformation of gray face images which are less variant to illumination changes when compared to the actual gray images. A measure namely $H_{pv}$ to compare the transformed face images based on partial Hausdorff distance has been defined and its time efficient implementation has been given. Comparison of the recognition method based on the proposed measure with some of the existing methods that work on edge maps of faces using benchmark faces varying in expression, illumination and slight pose shows that the proposed method outperforms the existing ones. This is due to appearance-based comparison of faces performed by the new measure as human does.

### References