Reducing Computational Complexity of New Modified Hausdorff Distance Method for Face Recognition Using Local Start Search

Chau Dang-Nguyen[1,2] and Tuan Do-Hong[1,2]
[1] Ho Chi Minh City University of Technology (HCMUT), District 10, Ho Chi Minh City, Vietnam
[2] Vietnam National University Ho Chi Minh City, Thu Duc District, Ho Chi Minh City, Vietnam
Email: {chaudn; do-hong}@hcmut.edu.vn

***Abstract*—Average Hausdorff distance that is an efficient measurement is widely used in face recognition method for measuring the dissimilarity between two sets of features. The New modified Hausdorff distance (MMHD) is a face recognition method, which uses average Hausdorff distance for measuring the dissimilarity between two sets of dominant points, which are features of face image. However, the disadvantage of the average Hausdorff distance is high computational complexity. Various methods have been proposed in recent decade with the purpose of reducing the complexity of Hausdorff distance computing. Local start search (LSS) is a state-of-art method for reducing the complexity of the Hausdorff distance computing. In this paper, we present how to use the LSS method for reducing the complexity of the computing the average Hausdorff distance. Firstly, a modification of the MMHD method, namely Least Trimmed New Modified Hausdorff distance (LT-MMHD) is proposed. The LT-MMHD method uses average Hausdorff distance of largest values for measuring the distance between two sets of dominant points. The proposed method gives higher recognition rate than the MMHD method for all conditions of face image. Finally, the LSS method is used for reducing the computational complexity of the proposed method. Experimental results show that by using the LSS method, the proposed method could reduce the computational complexity of 17%.**

***Index Terms*—Hausdorff distance, computational analysis, face recognition, Local start search, local feature, SSPP.**

I. INTRODUCTION

Face recognition is an important and active topic in computer vision and pattern recognition field due to its burgeoning applications in many critical areas such as access control, passport verification, surveillance and so on. Due to this, face recognition has attracted researchers from various fields such as image processing, machine learning, computer vision, etc. In recent year, a lot of face recognition methods have been proposed. However, various problems still challenge the researchers such as illumination, face expression, and pose [1]. One of the most important problem in face recognition filed is only single training sample per person [2]. In many applications, there is only one image for training or it is very difficult to collect multiple images of a person such as passport identification, credit card verification, driver's license identification, etc. Such problem is called single image face recognition (SSPP). The advantages of SSPP is low cost of collecting image for training and low storage cost [3], [4]. However, the recognition rate of face recognition methods decreases in the SSPP situation [5], [6].

Several face recognition methods have been proposed in last decade for solving the SSPP problem. These methods could be divided into five categories: global feature-based methods [7]-[9], local feature-based methods [10]-[13], generic database-based methods [14]-[17], virtual sample generation methods [18]-[20] and hybrid methods [21], [22]. In comparing with other methods in other categories, the face recognition methods based on local feature are easy implemented in real applications [1], invariant with non-ideal conditions of face image such as lighting, orientation or noise [2].

Edge is a local feature widely used in face recognition methods because of its invariant characteristic with non-ideal lighting condition of face image [23]. Various face recognition methods use edge pixels as face image feature [24]-[30]. The New Modified Hausdorff distance method [31] is a face recognition method that uses dominant points of edge pixel as the features of face image. In comparing with other methods using edge pixels as feature of face image, the MMHD method has much lower storage cost, which has 80% lower storage cost. These methods [24]-[31] use average Hausdorff distance for measuring the dissimilarity between two sets of edge pixels or two sets of dominant points. The average Hausdorff distance is the average of all distances between pairs of corresponding points in two sets. The disadvantage of average Hausdorff distance is high computational complexity. Thus, the computational complexities of the face recognition methods using average Hausdorff distance are very high. This prevents the methods that used average Hausdorff distance from real face recognition applications.

Manuscript received October 12, 2020; revised January 12, 2021; accepted January 22, 2020.

Corresponding author: Dang Nguyen Chau (email: chaudn@hcmut.edu.vn).

This research is funded by Ho Chi Minh City University of Technology (HCMUT), VNU-HCM under grant number HCMUT-003260-2021.

doi: 10.18178/ijeetc.10.4.261-271

Hausdorff Distance (HD), or MAX-MIN distance, is a useful measurement to determine the degree of resemblance between two point sets. Hausdorff distance is widely used in many domains such as pattern recognition, shape matching, segmentation techniques for medical image, CAD/CAM, etc. However, the computing of the HD is high complexity because it contains both maximization and minimization rather than just one or the other. In recent decade, various methods had been proposed with the purpose reducing the computational complexity of computing the HD [32]-[37]. However, these methods cannot be used for reducing the complexity of the average HD computing.

In this paper, the state-of-art Local Start Search (LSS) method [35] is applied for reducing the computational complexity of the proposed method, which is based on the average Hausdorff distance. Firstly, a modification of the MMHD method, the Least Trimmed New Modified Hausdorff distance (LT-MMHD) method for face recognition is proposed. The LT-MMHD method uses dominant points set as face image feature, the same as the MMHD method. However, it is different from the MMHD method, the LT-MMHD method uses the least trimmed average Hausdorff distance, which is the average of the largest distances between pairs of corresponding points in two sets, for measuring the dissimilarity between two sets. After that, the LSS method is used for reducing the computational complexity of the proposed method, the LT-MMHD method. The experimental results show that the LT-MMHD method gives higher recognition rate than MMHD in all experiments. Furthermore, the LT-MMHD method has 17% lower computational complexity than the MMHD method.

In the following, a brief review of the methods for reducing the computation complexity of HD computing is given in Section II. In Section III, the proposed method, the LT-MMHD method, is presented. The experimental results are shown in Section IV. Finally, the paper is concluded in Section V.

II. Related Work

Given two point sets $M=\{m_1,m_2,\cdots,m_P\}$ and $T=\{t_1,t_2,\cdots,t_Q\}$, where P and Q are the number of points in set M and T, respectively. The directed Hausdorff distance $h(M,T)$ from M to T is the maximum distance of a point $m\in M$ to its nearest neighbor $t\in T$. The directed HD from M to T as a mathematical formula is

$$h(M,T)=\max_{m\in M}\min_{t\in T}\|m-t\| \quad (1)$$

where $\|\cdot\|$ is any norm distance metric, e.g. the Euclidean distance. For each point $m\in M$ has to calculate the Euclidean distance to all points $t\in T$ for finding the minimum distance. The computational complexity of this task is $O(Q)$. This task must be done with all points $m\in M$. So, the complexity of the HD computing as (1) is $O(PQ)$. Note that, in general case, $h(M,T)\neq h(T,M)$ and thus directed HD is not symmetric. The HD between M and T is defined as the maximum of both directed HD and thus it is symmetric. The HD $H(M,T)$ is defined as

$$H(M,T)=\max\left(h(M,T),h(T,M)\right) \quad (2)$$

Computing HD is challenging because its characteristic contains both maximization and minimization. Many efficient algorithms, in recent decades, have been proposed for reducing the computational complexity of the HD computing. Depending on the data type of two sets, these methods could be divided into three categories: 1) the efficiency methods for computing the HD between two polygonal models, 2) the efficiency methods for computing the HD between two curves or mesh surfaces and 3) the efficiency methods for computing the HD between two point sets.

However, the methods for efficiency HD computing between two polygonal models, two curves or two mesh surfaces are not general because these algorithms based on specific of data types [36]. The methods for efficiency HD computing between two point sets are more general and can be generally divided into two categories, 1) approximation HD and 2) exact HD. In the first category, which is approximation of HD, the algorithms try to efficiently find an approximation of the HD. These algorithms have been widely used in runtime-critical applications. On the other hand, the algorithms of the second category aim to efficiently compute the exact HD. The surveys in [38], [39] presented more details about these methods in the first category. In the face recognition field, exact HD is widely used for measuring the distance between two set of features. In this paper, we present some general methods for efficiency exact HD computing between two point sets.

In [32], an algorithm for finding the Aggregate Nearest Neighbor (ANN) in database was presented. This algorithm uses the R-Tree for optimizing the searching for ANN. As the development of the algorithm in [32], the method in [33] presented two algorithms, the Depth-First Hausdorff distance algorithm (DF-HD) and the Best-First Hausdorff distance algorithm (BF-HD). These methods use two R-tree structures for two point sets for optimizing the nearest neighbor searching. By using the index and the bounding distance, both of algorithms, the DF-HD and the BF-HD, quickly prune the non-contributed nodes for final HD computation. Based on DF-HD and BF-HD, an efficient method, Incremental Hausdorff Distance (IHD), for HD calculation between two point sets was proposed. The algorithm uses two R-Trees at the same time, each R-tree for each point set, to avoid the interaction of all points in two sets. The aggregate nearest neighbor is simultaneously determined in both directions. However, complex structure of above algorithms makes the computation cost increase and the R-Tree structure is not suitable for general point sets.

A fast and efficient algorithm for computing exact HD between two point sets, the EARLYBREAK method, was proposed in [34]. The algorithm has two loops, with the outer loop for maximization and the inner loop for minimization. If the distance between the current point in the outer loop and a point in the inner loop that is below the temporary HD, which could be named as $c_{\max}$, the minimum value of the distances between the current point in the outer loop to all points in the inner loop will be below $c_{\max}$. The distance from such point in the outer loop to its nearest neighbor does not make the value of HD change. Such point in the outer loop is called non-contributed point for the HD computing. The computing can break the inner loop as soon as a distance that is below $c_{\max}$ is found and the outer loop could continue with the next point. Moreover, for improving performance, random sampling was also presented in this algorithm to avoid similar distances in successive iterations.

Based on the EARLYBREAK method, a state-of-art method, namely Local Start Search (LSS), was presented in [35] for efficient computing exact HD between two arbitrary point sets. In the LSS method, the point set is ordered by using the Morton code, which is also known as the Morton curve or the Z-order. The main idea of the LSS algorithm is that if the break occurs in current loop at the point *x*, it is quite possible that the break will occur at the position near *x* in the next loop. Moreover, if the nearest neighbor at the point *x* in current loop, the nearest neighbor in next loop is also quite possible at the point near *x*. In LSS algorithm, a variable, namely *preindex*, is used for preserving the location of break occurrence or the location of the nearest neighbor point in current outer loop. In the next outer loop, the inner loop will start from the *preindex* and scan its neighbor for both sides of the *preindex*.

For 3D point set, a method for computing the Hausdorff distance between two general 3D point sets was presented in [36]. This method contains two sub-algorithms Nonoverlap Hausdorff Distance (NOHD) and Overlap Hausdorff Distance (OHD). In [37], a framework of diffusion search was presented for efficient and accurate HD computing between two 3D models. The method uses two algorithms, the LSS algorithm for computing the HD between two sparse 3D point sets and the OHD algorithm for HD computing between two dense 3D point sets.

In this paper, the LT-MMHD method uses the average HD for measuring the dissimilarity between two sets of dominant points, which are the 2D point sets. The methods in [36] and [37] are used just for 3D point sets HD computing and are not suitable for the 2D point sets HD computing. And thus, the LSS method is used for reducing the computational complexity of the LT-MMHD method.

III. Proposed Method

A. MMHD Method

An edge presents a rapid change in gray level of a face image, which reflects the geometrical structure of human face. Edge is used as an important feature for face recognition. An edge detector followed by a thinning process is applied to face image for generating one-pixel thick edge. The Dyn2S algorithm [40] is applied to binary edge image for removing edge pixels which have low curvature. The remaining edge pixels of binary edge image are called dominant points.

Given two finite dominant point sets $M = \{m_1, m_2, \cdots, m_P\}$ (representing the model in the database) and $T = \{t_1, t_2, \cdots, t_Q\}$ (representing the test image), the HD of the MMHD method between two sets is defined as

$$H_{\text{MMHD}}(M,T) = \max\left(h_{\text{MMHD}}(M,T), h_{\text{MMHD}}(T,M)\right) \quad (3)$$

where $h(M,T)$ is the directed HD from *M* to *T* and $h(T,M)$ is the directed HD from *T* to *M*. The directed HD $h(M,T)$ is defined as

$$h_{\text{MMHD}}(M,T) = \frac{1}{\sum_{m_i \in M} W_{m_i t_j}} \sum_{m_i \in M} W_{m_i t_j} \cdot \min_{t_j \in T} \left\| m_i - t_j \right\| \quad (4)$$

where $W_{m_i t_j} = 1/2(W_{m_i} + W_{t_j})$ is the average merit of the dominant point m_i and t_j ; W_{m_i} and W_{t_j} are the merit provided by the Dyn2S algorithm previously mentioned; $\| m_i - t_j \|$ is the Euclidean distance between two dominant points m_i and t_j.

B. Least Trimmed New Modified Hausdorff Distance (LT-MMHD) for Face Recognition

Supporting $M = \{m_1, m_2, \cdots, m_P\}$ and $T = \{t_1, t_2, \cdots, t_Q\}$ are the dominant point sets of the model image and the test image, respectively. Here, we proposed the Least Trimmed New Modified Hausdorff distance (LT-MMHD) for measuring the dissimilarity between two dominant point sets. The directed Hausdorff distance of the LT-MMHD method from *M* to *T* is defined as

$$h_{\text{LT-MMHD}}(M,T) = \frac{1}{\sum_{i=K}^{P} W_{m_i t_j}} \sum_{i=K}^{P} W_{m_i t_i} \left(\min_{t_j \in T} \left\| m - t_j \right\| \right)_i \quad (5)$$

where $(\min_{t_j \in T} \| m - t_j \|)_i$ represents the *i*th value in the sorted sequence in ascending order $(\min_{t_j \in T} \| m - t_j \|)_{m \in M}$; *P* is the number of dominant points in the model image and *K* is a parameter that is defined as

$$K = f \times P \quad (6)$$

which f is a given fraction, that determines the performance of the proposed method.

As (4), the directed HD of the MMHD method from set M to set T is the average of all distances from all points $m \in M$ to their nearest neighbors. Different from the directed HD of the MMHD method, the directed HD from set M to set T of the LT-MMHD method as in (5) is the average of the largest distances, which are greater or equal to the K ranked value, from a point $m \in M$ to its nearest neighbor $t \in T$. A primary HD of the LT-MMHD (pLT-MMHD) method has mathematical formula as below:

$$H_{\text{pLT-MMHD}}(M,T) = \max\left(h_{\text{LT-MMHD}}(M,T), h_{\text{LT-MMHD}}(T,M)\right) \quad (7)$$

However, the design of pLT-MMHD still has the following weakness. Supporting T is the set of dominant points of test image and t is a point in the set T. M_c and M_n are the dominant point sets of corresponding image and non-corresponding image, respectively, for the test image in the database. Due to the error in feature extraction process, the corresponding point m_c of the point t in the corresponding set M_c is missing, the point t will take another point in set M_c as its nearest neighbor, the point m_{cn}. As (5), the distance $\|m_{\text{cn}} - t\|$ is used for calculating the directed HD of the pLT-MMHD from T to M_c. Similarly, the point t takes the point $m_n \in M_n$ as its nearest neighbor. The distance $\| m_n - t \|$ is used for calculating the directed distance from T to M_n. It is possible to have the value of $\| m_{\text{cn}} - t \|$ is much higher than that of $\| m_n - t \|$, and as in (5), the distance between T and M_c is much higher than the distance between T and M_n. The test image will take the non-corresponding image as the matching image. This is an undesired mismatch.

The number of high confident corresponding pairs between the model image and the test image should be used as a measure of similarity. A point $m \in M$ finds that a point $t \in T$ is its nearest neighbor and the distance between them is lower than a given distance N_p, then such point m is called as a *high confident point*. The number of high confident points between the test image and its corresponding image is much higher than between the test image and its non-corresponding image. A *high confident ratio* of an image is defined as the ratio between the number of high confident points (N_{hc}) to the number of total dominant points (N_{total}).

$$R = N_{\text{hc}} / N_{\text{total}} \quad (8)$$

The disparity number between two images is defined as below

$$D_n = 1 - \frac{R_M + R_T}{2} \quad (9)$$

where R_M and R_T are the high confident ratio of the model image and the test images, respectively. The complete version of Hausdorff distance between two sets of the LT-MMHD method is defined as

$$H_{\text{LT-MMHD}}(M,T) = \sqrt{H^2_{\text{pLT-MMHD}}(M,T) + (W_n D_n)^2} \quad (10)$$

where W_n is the weight of the disparity number and it is determined by a training process.

By using face database of BERN university [42] for training process, we obtain that: N_p=6 and W_n=25.

C. Local Start Search (LSS) for LT-MMHD

The directed distance of the LT-MMHD method as in (5) is the average of (P–K) largest distances from a point m to its nearest neighbor. Assuming that (P–K) temporary largest distances is found, and the minimum value of those distances is called as $c_{\max}$. For computing the distance from a point m to its nearest neighbor, the distances from the point m to all points $t \in T$ must be calculated and the minimum distance can be found from these distances, we call this is the inner loop. The key of reducing the running time of the LT-MMHD method is to reduce the average number of iterations of the inner loop. If the point m in the outer loop has the distance to a point t in the inner loop that is below the $c_{\max}$, such point m is called non-contributed for computing the HD between two dominant point sets. Therefore, the calculating distances from the point m in the outer loop to the remaining points in the inner loop is not necessary. The algorithm could break and continue to the next point of the outer loop as soon as a point t in the inner that makes the distance from it to the current point m in the outer loop below $c_{\max}$. If with the point m in the outer loop, the algorithm breaks at the position X in the inner loop, it is quite possible that with the next point of m in the outer loop, the algorithm could break at the point near X. Moreover, if X is the place of the nearest neighbor of the point m in the outer loop, the next point in the outer loop quite finds its nearest neighbor near X. For obtaining better locality of the dominant points in the set, the Morton code is used for ordering points.

The Morton code was first introduced by Guy Macdonald Morton [39] in 1966. The Morton code is a function that maps multidimensional data to one dimension while preserving the locality of the data point. In this study, the Morton code is used for ordering the dominant points in set. As shown in Table I, each dimension is encoded with k bits, the Morton code will have $2k$ bits. Fig. 1 shows the order of points along the Morton-curve.

TABLE I: ILLUSTRATION OF MORTON CODE

Unordered index	Coordinate	Morton code	Ordered index
a	(1,1)	0000	1
b	(1,2)	0001	2
c	(1,3)	0100	5
d	(1,4)	0101	6
e	(2,1)	0010	3
f	(2,2)	0011	4
g	(2,3)	0110	7
h	(2,4)	0111	8
i	(3,1)	1000	9
J	(3,2)	1001	10

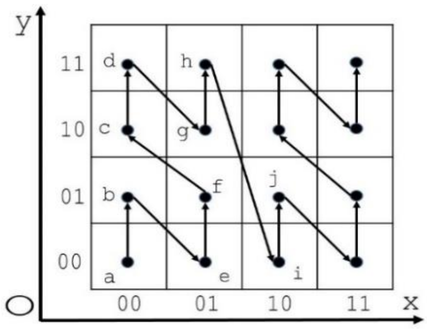

Fig. 1. Dominant point set ordered by Morton code.

Algorithm 1 describes our proposed method, using LSS for reducing the complexity of computing the directed HD of the LT-MMHD method. Line 9 and line 12 are called outer loop and inner loop, respectively. The function DIST(,) in Algorithm 1 denotes the Euclidean distance between two points; the symbol $\langle \cdot,\cdot \rangle$ denotes the inner product between two vectors.

Algorithm 1 LSS for computing the directed LT-MMHD

```
1:  Input: Edge map M and T, fraction f
2:  Output: Directed Hausdorff distance H_MT
3:  Create the Z-order point set M^z
4:  Create the Z-order point set T^z
5:  rank = (1 – f) * length(M^z);
6:  hMT = zeros(2,rank);
7:  preindex = round(length(T^z)/2);
8:  minplace = 0;
9:  for i = 1:length(M^z) do
10:   [c_max, place_hMT] = min(hMT(1,:));
11:   c_min = inf;
12:   for j = 0:length(T^z) do
13:     u = preindex – j;
14:     v = preindex + j;
15:     if u >= 1 then
16:       d_left = DIST(M^z(i),T^z(u));
17:       if d_left < c_max then
18:         preindex = u;
19:         c_min = 0;
20:         break
21:       endif
22:       if d_left < c_min then
23:         c_min = d_left;
24:         minplace = u;
25:         W = W(T^z(u));
26:       endif
27:     endif
28:     if v <= length(T^z) then
29:       d_right = DIST(M^z(i),T^z(v));
30:       if d_right < c_max then
31:         preindex = v;
32:         c_min = 0;
33:         break
34:       endif
35:       if d_right < c_min then
36:         c_min = d_right;
37:         minplace = v;
38:         W = W(T^z(v));
39:       endif
40:     endif
41:   endfor
42:   if c_max < c_min then
43:     hMT(1,place_hMT) = c_min;
44:     hMT(2,place_hMT) = 0.5*(W+W(M^z(i)));
45:     preindex = minplace;
46:   endif
47: endfor
48: return H_MT = ⟨hMT(1,:),hMT(2,:)⟩/sum(hMT(2,:));
```

The main steps of Algorithm 1 are summarized as follows.

- Create the Z-order of two dominant point sets. Initialize a matrix *hMT* for saving the $(P–K)$ temporary largest distances from a point m in the outer loop to its nearest neighbor $t \in T$ in the inner loop $\| m - t \|$. The variable $c_{\max}$ is assigned as the minimum value of these distances. The variable *preindex* is used as the starting location of the inner loop. The variable *minplace* is assigned as the location of the nearest neighbor of the previous point in the outer loop.
- The outer loop traverses all points in M^z. A variable, namely $c_{\min}$, is used for preserving the minimum value of the distances from the current point in the outer loop to points in the inner loop.
- The inner loop executes a searching on T^z at the starting location *preindex*. Both forward and backward searches are performed as the same time on T^z.
- If a distance d between the point t in the inner loop to the current point in the outer loop is below the value $c_{\max}$, the algorithm will break the inner loop and continue with next point in the outer loop. The place that the break occurs is used for updating the value of *preindex*. If the break does not occur, the value of d is used for updating $c_{\min}$. The starting location of inner loop *preindex* is also updated by using the variable *minplace*.
- The value of $c_{\min}$ is used for updating the matrix hMT.

It is noted that the algorithm does not break during first $(1-f)P$ iterations of the outer loop, where P is the number of dominant points of the model image, because the value of $c_{\max}$ is equal to zero during these iterations, and the distance between two points d cannot be below the value of $c_{\max}$.

D. Computational Complexity Analysis

Supporting P and Q are the number of points in the point sets M and T, respectively. For calculating the directed Hausdorff distance of the MMHD method as in (5), it is required to calculate distances from each point $m \in M$ to all points $t \in T$ for finding the minimum value, which is the distance from m to its nearest neighbor. This task must be performed for all points $m \in M$. The directed Hausdorff distance of the MMHD method is the average of all distances from m to its nearest neighbor. The complexity of the directed Hausdorff distance MMHD computing is $O(PQ)$.

The Algorithm 1 has computational complexity is $O((1–f)PQ)$ for the best case, where the algorithm breaks at the first iteration of the inner loop. For the worst case, the break does not occur for all iterations, the computational complexity is $O(PQ)$. Hence, the computational complexity of the MMHD method is the lower bound of the computational complexity of our proposed method. In general case, the computational complexity of the proposed method is $O((1–f)PQ+fPR)$, where R denotes the average number of iterations in the inner loop. The lower value of R, the better of performance of the proposed method. Now let us discuss the value of R in general case through the analysis of probability theory.

In the Algorithm 1, considering that a random point $t \in T$ in the inner loop, and finding a distance d between point t to the current point m in the outer loop. We define the event e to be the non-appearance of the break in the algorithm, or the event of that finding distance d larger than $c_{\max}$, which is the minimum value of the temporary largest distances from a point m in the outer loop to its nearest neighbor $t \in T$ in the inner loop as above mentioned. Assuming that the event *e* occurs with the probability *P*(*e*)=*q*. Obviously, the event $\bar{e}$ means the appearance of the break in the algorithm and the probability of this event is $P(\bar{e}) = 1 - q = p$.

We also define a random variable *R* as the number of successive the event $\bar{e}$ until the event *e* occurs. The variable *R* also means the average number of iterations before an early break occurs. This is equivalent to $R-1$ distances from the current point of the outer loop *m* to the points $t_1, t_2, \cdots, t_{R-1}$ namely $d_1, d_2, \cdots, d_{R-1}$ are higher than $c_{\max}$ and one distance d_R is below $c_{\max}$. The probability density function of random variable *R* is given by

$$\begin{aligned} f(r) &= P(d_1 > c_{\max}, \cdots, d_{r-1} > c_{\max}, d_r \le c_{\max}) \\ &= q \times \cdots \times q \times p = q^{r-1} p \end{aligned} \tag{11}$$

The expectation of average number of iterations until an early break occurs equivalent to the expected value of *f*(*r*).

$$E[R] = \sum_{r=1}^{\infty} r f(r) = \sum_{r=1}^{\infty} r q^{r-1} p \tag{12}$$

Then, equation (12) can be rewritten in the form of a polynomial as follow

$$E[R] = p + 2qp + 3q^2 p + 4q^3 p + \cdots \tag{13}$$

A simpler formula of this sum could be found by multiplying both side with *q*, subtracting the resulting equation from (12) and then dividing by *p*

$$\frac{E[R](1-q)}{p} = 1 + q + q^2 + q^3 + \cdots = \frac{1}{1-q} \tag{14}$$

Obviously, by using *q*=1–*p* to (12), the convenient formula is given as

$$E[R] = \frac{1}{p} \tag{15}$$

Equation (15) tells the important fact that the number of iterations until a break occurs only depends on *p*, which is the probability of finding a distance between two points below the value of $c_{\max}$. The higher of *p* is, the lower of number iterations in the inner loop, and vice versa. Thus, the high probability *p*, we can reduce the average number of iterations in the inner loop. But the question is how high is *p* and what does it depend on?

In fact, *p* depends mainly on the value of $c_{\max}$. The larger $c_{\max}$ is, the easier to find a point in the inner loop that makes the distance to the current point of outer loop below $c_{\max}$, which means the higher value of probability *p* and vice versa. Fig. 2 illustrates the relation between $c_{\max}$, probability *p* and the distribution of pairwise distance between two points. Here, the normal distribution is used for illustration. The example in Fig. 2 shows that the value of *p* does not depend on the size of point set, but rather on the value of $c_{\max}$ and the distribution of the pairwise distance.

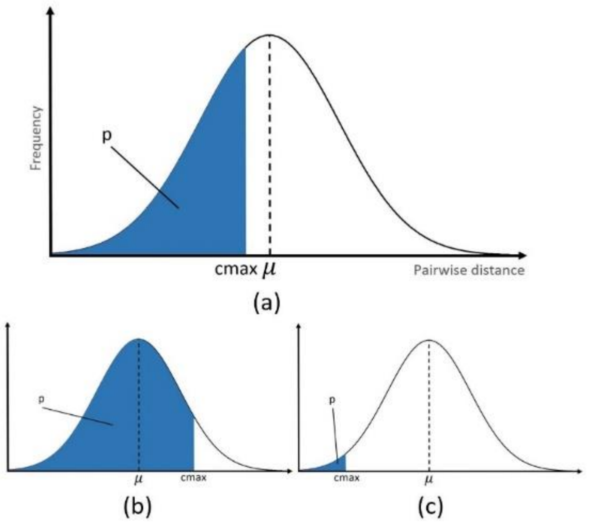

Fig. 2. Distribution of pairwise distance assuming a normal distribution for illustration: (a) Relation between $c_{\max}$ and probability *p*, (b) Large value of $c_{\max}$ thereby increasing *p*, and (c) Small value of $c_{\max}$ thereby decreasing *p*.

The value of $c_{\max}$ is determined by the fraction *f*. The larger of *f* is, the lower of number of largest distances are used for computing directed distance of the LT-MMHD method as in (5) and (6), which means the higher value of $c_{\max}$ and vice versa.

IV. Experimental Results

In this section, the performance of the proposed method, the LT-MMHD method, is evaluated for face recognition application. The recognition rate, which is the ratio of number of images correctly classified to the total number of images in the test set, is used for evaluating. The recognition rate of the proposed method is also compared with the MMHD method that uses average Hausdorff distance.

In this study, the face database from the University of Bern [42] and the AR face database from Purdue University [43] are used. Bern university face database contains frontal views of 30 people. Each person has 10 gray images with different head pose variations: two frontal pose images, two looking to the right images, two looking to the left images, two looking upward image and two looking downward images. The AR face database contains 2599 color face images of 100 people (50 men and 50 women), there are 26 images for each person and be divided into 2 sessions separated by two-week interval. Each session has 13 images which are the frontal view faces with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf). However, one of frontal face image is corrupted (W-027-14.bmp) and only 99 pairs of face image are used for examining the performance of system for face recognition under normal conditions. In our experiment, a preprocessing before recognition process is used for locating the face. All images are normalized such that the two eyes are aligned roughly at the same position with a distance of 80 pixels. After that, all images are cropped with size 160×160 pixels. The experiments are conducted on the PC with 2.5GHz CPU and 4GHz RAM.

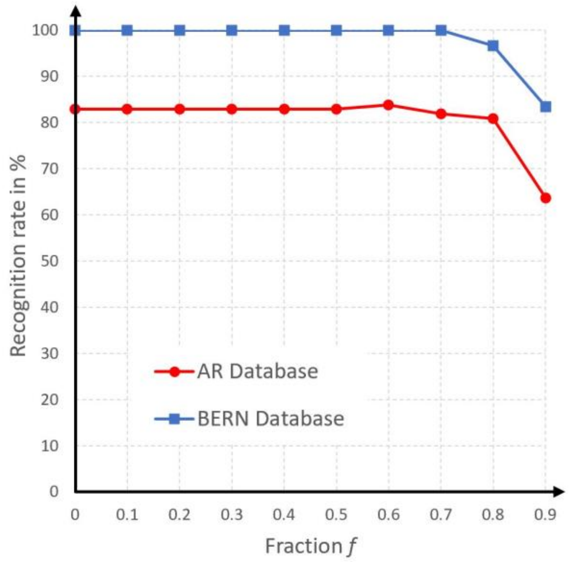

Fig. 3. The influence of fraction *f* on the recognition rate of the LT-MMHD method with the Bern and AR database.

A. Influence of Fraction f on the Perfomance of the Proposed Method

In the proposed method, fraction f is used for determining the number of dominant points which are used for calculating the directed distance of the LT-MMHD method as in (3). If the value of f is high, a low number of dominant points are used for calculating the directed distance of the LT-MMHD method. The outliers are commonly the points that have the very large distance to its nearest neighbor. This means, if the value of f is high, most of dominant points used for calculating the directed distance of the LT-MMHD method are the outliers. And thus, the recognition rate of proposed method is very low. On the other hand, if the value of f is too low, high number of dominant points in the set are used for calculating the directed distance of the LT-MMHD method. This means not only the dominant points in discriminated regions of faces but also the dominant points in similar regions of faces are used for calculating. And thus, the contribution of dominant points in discriminated regions of faces become low and the recognition rate of proposed method might slightly decrease.

Fig. 3 shows the recognition rate of the proposed method with various value of fraction f using the Bern database and the AR database with frontal face images. The value of fraction f is changed from 0 to 0.9. The value f of 1 is meaningless because there are no dominant points used for calculating the directed distance of the LT-MMHD method. As the result shown in Fig. 3, the recognition rate of the proposed method reaches the highest value at $f = 0.6$ for the AR database and at $f \leq 0.7$ for the BERN database.

Beside recognition rate, the fraction f also influences the computational complexity of the proposed method. The low value of f means the longer size of matrix hMT in the Algorithm 1, and thus, the lower value of c_{max}. The low value of c_{max} makes the average number of iterations in the inner loop increase, and also makes the computation cost of proposed method increase. The value of fraction f is chosen at 0.6 for following experiments in this section.

The complexity of the computing directed distance of MMHD is $O(PQ)$. With the value 0.6 of fraction f, the proposed method has the computational complexity of $O(P(0.4Q+0.6R))$, where R is the average number of iterations in the inner loop. As in [36], the value of R is lower than Q. And thus, the proposed method has lower computational complexity than the one of the MMHD method. Table II shows the average number of iterations in the inner loop when computing directed Hausdorff distances of the MMHD method and the LT-MMHD method with a pair of face image in the AR database and the database of BERN University. The proposed method has approximately 17% lower average number of iterations in the inner loop and it has also lower computational complexity, than the MMHD method.

B. Face Recognition under Normal Condition

The frontal face images in normal conditions in the BERN university database and the AR database are used for evaluating the performance of the proposed method. Each person has two images, one for test set and one for the model set. The recognition rates of different methods are given in Table III. The recognition rates of all methods with BERN university database are higher than those with the AR database because two reasons. The first reason is the difference between a pair image of each person in the AR database is larger than the BERN database. And the second reason is the illumination of model image and test image in AR database is also different. The proposed method gives 7% higher recognition rate with the BERN database and 10 % higher recognition rate with the AR database than the MMHD method.

Moreover, in the case $f = 0$, the LT-MMHD method is equal to adding the disparity number as in (7) to the MMHD method. As the result in Table III, by adding the disparity number, the recognition rate of the MMHD method will increase. However, the recognition rate of the LT-MMHD method is 1% higher than the recognition rate of the MMHD method with the disparity number in the case of the AR database. This means that the average Hausdorff distance of largest values gives better performance than the average Hausdorff distance.

TABLE II: NUMBER OF AVERAGE INNER ITERATIONS OF DIRECTED HAUSDORFF DISTANCE BETWEEN PAIR IMAGE USING THE BERN AND AR DATABASES

Method	BERN database		AR database	
	MMHD	LT-MMHD	MMHD	LT-MMHD
Average number of inner loop ($\times 10^3$)	1204	1003	792.9	660.8
Decrease (%)	-	**16.67**	-	**16.66**

TABLE III: RECOGNITION RATE IN NORMAL CONDITION

Method	MMHD	MMHD with disparity number	LT-MMHD
Bern database	93.33%	100%	**100%**
AR database	74.75%	82.83%	**83.84 %**

TABLE IV: RECOGNITION RATE FOR DIFFERENT LIGHTING CONDITIONS

Method	MMHD	MMHD with disparity number	LT-MMHD
Left light on	71%	81%	**85%**
Right light on	64%	74%	**80 %**
Both light on	49%	61%	**76%**
Average	61.33%	72%	**80.33%**

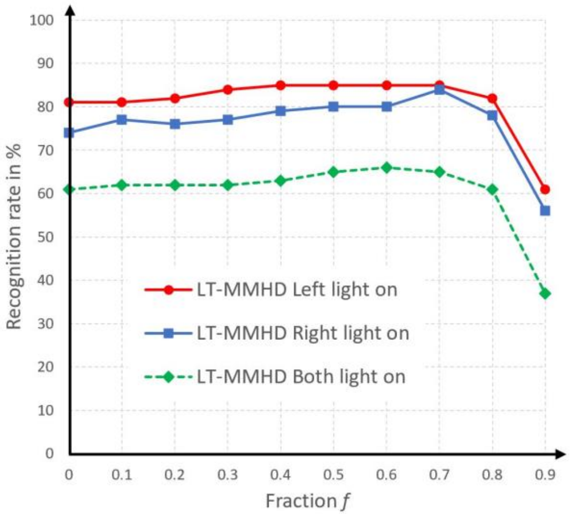

Fig. 4. The influence of fraction *f* on the recognition rate of the LT-MMHD method with under varying lighting conditions.

C. Face Recognition under Varying Lighting Conditions

The performance of the proposed method is also compared with the MMHD method for face recognition under varying lighting conditions of face image. For evaluating the performance of the methods under varying lighting conditions of face image, the images in the AR database are used. Frontal face images in normal lighting condition for 100 people are used as model set. The face images with a light source on the left side of face, a light source on the right side of face and two light sources on both sides of face, are divided into three test sets with 100 images for each set. The recognition rates of the methods are given in Table IV.

The proposed method gives higher recognition rate of approximate 19% than the MMHD method for different lighting conditions of face image on average. Especially, the LT-MMHD method gives higher recognition rate of approximate 8% than the MMHD method with disparity number.

Fig. 4 describes more detail about the influence of fraction f on the recognition rate of the LT-MMHD method for different lighting conditions. With the condition of left light on, the recognition rate of the LT-MMHD method reaches the maximum value at $f = 0.6$ and 4% higher than the recognition rate at $f = 0$. With the condition of right light on, the recognition rate of the LT-MMHD method reaches the maximum value at $f = 0.6$ and 5% higher than the recognition rate at $f = 0$. Especially, with condition of two light sources on both sides of face, the recognition rate of the LT-MMHD method reaches the maximum value at $f = 0.7$, 4% higher than the recognition rate at $f = 0.6$ and 10% higher than that at $f = 0$. This means that the average Hausdorff distance of largest values gives better performance than the average Hausdorff distance.

TABLE V: RECOGNITION RATE FOR DIFFERENT FACE EXPRESSIONS

Method	MMHD	MMHD with disparity number	LT-MMHD
Smile	34%	35%	**76%**
Angry	50%	62%	**84%**
Average	42%	48.5%	**80%**

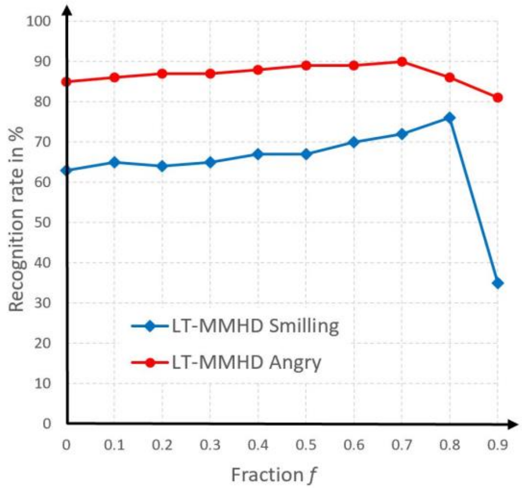

Fig. 5. The influence of fraction *f* on the recognition rate of the LT-MMHD method with different face expression.

D. Face Recognition for Different Face Expressions

For evaluating the performance of the methods with different face expressions, the face images with different face expressions in the AR are used. The face images with two face expressions, smile and angry, are divided into two test sets with 100 images for each set. The frontal face images with neutral expression of 100 people are used as the model set. The Table V shows the recognition rates of the methods.

The proposed method gives higher recognition rate of 38% than the one of the MMHD method for different face expressions, on average. The proposed method also gives higher recognition rate of 31% than MMHD method with disparity number.

Fig. 5 describes more detail about the influence of fraction f on the recognition rate of the LT-MMHD method for different face expressions. With the smiling face image, the recognition rate the LT-MMHD method reaches the maximum value at $f = 0.7$, 1% higher than the recognition rate at $f = 0.6$ and 5% higher than that at $f = 0$. Especially, with the angry face image, the recognition rate of the LT-MMHD method reaches the maximum value at $f = 0.8$, 6% higher than the recognition rate at $f = 0.6$ and up to 13% higher than that at $f = 0$. In the case of face recognition with different face expressions, the average Hausdorff distance of largest values gives much better performance than the average Hausdorff distance.

E. Face Recognition for Different Face Poses

The face database of BERN University is used for evaluating the performance of the methods with different face poses. The test sets contain images of 30 peoples with different poses, e.g. looking to the left and right, looking up and looking down. The model set contains 30 frontal face images of 30 people. The recognition rates of the methods for different face poses are presented in Table VI.

TABLE VI: RECOGNITION RATE FOR DIFFERENT FACE POSES

Method	MMHD	MMHD with number disparity	LT-MMHD
Looks left	40%	53.33%	**53.33%**
Looks right	53.33%	63.33%	**66.67%**
Looks up	60%	56.67%	**60%**
Looks down	53.33%	63.33%	**66.67%**
Average	51.67	59.17%	**61.67%**

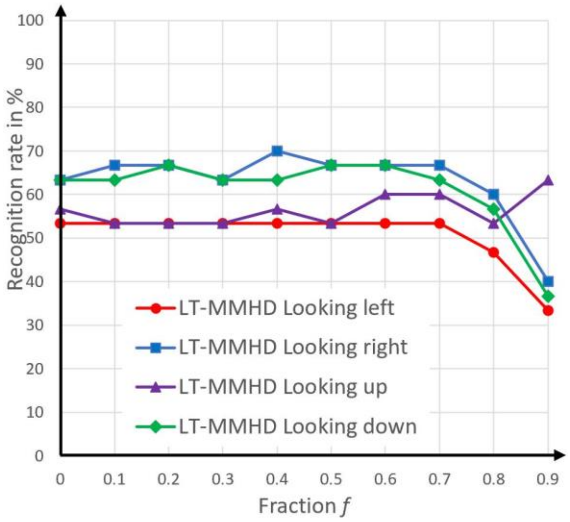

Fig. 6. The influence of fraction *f* on the recognition rate of the LT-MMHD method with different face poses.

On average, the proposed method gives 10% higher recognition rate than the MMHD method. In comparing with the MMHD method with disparity number, the LT-MMHD method also gives 2% higher recognition rate on the average.

Fig. 6 describes more detail about the influence of fraction f on the recognition rate of the LT-MMHD method with different face poses. With the face looking left, the recognition rate of the LT-MMHD method reaches the maximum value at $f = 0.7$ and it equals to that at $f = 0$. With the face looking right, the recognition rate of the LT-MMHD method reaches the maximum value at $f = 0.4$, 3% higher than the recognition rate at $f = 0.6$ and 7% higher than that at $f = 0$. With the condition of the face looking up and looking down the recognition rate of the LT-MMHD method reaches the maximum value at $f = 0.6$ and 3% higher than that at $f = 0$. In the case of face recognition with different face poses, the average Hausdorff distance of largest values gives higher performance than the average Hausdorff distance.

V. CONCLUSION

Face recognition is an important research due to its burgeoning applications in modern life. In last three decades, a lot of face recognition methods have been proposed. However, the SSPP problem, which has only one image for each person in the database, still challenges the researchers in computer vision and pattern recognition fields. In last decade, several face recognition methods have been proposed for solving the SSPP problem. These methods could be divided into five categories: global feature-based methods, local feature-based methods, virtual image generation methods, generic database-based methods and the hybrid methods. In comparing with other methods, the methods in local feature-based category are simple and easy to deploy in real face recognition applications.

Edge is an important local feature, which is variant with non-ideal conditions of face image, especially under varying lighting conditions of face image. Various face recognition methods for solving the SSPP problem uses edge pixels as the features of face image. These methods use average Hausdorff distance for measuring the dissimilarity between two sets of features of two face images. However, the average Hausdorff distance has very high complexity of computing. And thus, the computational complexities of the face recognition methods use average Hausdorff distance are very high.

The MMHD method is a face recognition method use features of face image are the dominant points of edge pixels of face image. This method has very low storage cost in comparing with the face recognition methods use edge pixels as features of face image. However, the disadvantage of the MMHD method is high computational complexity because the average Hausdorff distance is used in the MMHD method for measuring the distance between two sets of dominant points.

In this paper, a modification the MMHD method, the LT-MMHD method for face recognition, was proposed. The LT-MMHD method also uses the set of dominant points as feature of face image, the same as the MMHD method. However, different from the MMHD method, the LT-MMHD method uses the average Hausdorff distance of largest values for measuring the distance between two sets of dominant points. The experimental results show that, the average Hausdorff distance of largest values gives better recognition rate than the average Hausdorff distance in all conditions of face image. This means replacing the average Hausdorff distance by the average Hausdorff distance of largest values leads to the improvement of recognition rates of face recognition methods using average Hausdorff distance.

Moreover, in this paper, the LSS method, a state-of-art method for reducing the complexity of the Hausdorff distance computing, was used for reducing the computational complexity of the LT-MMHD method. The experimental result shows that the computational complexity of the LT-MMHD method is approximate 17% lower than that of the MMHD method.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Chau Dang-Nguyen and Tuan Do-Hong contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript; all authors had approved the final version.

REFERENCES

[1] Y. Kortli, M. Jridi, A. A. Falou, and M. Atri, "Face recognition systems: A survey," *Sensors*, vol. 20, no. 2, p. 342, Jan. 2020.

[2] N. Kumar and V. Garg, “Single sample face recognition in the last decade: A survey,” *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 13, Apr. 2019.

[3] S. I. Choi, Y. Lee, and M. Lee, “Face recognition in SSPP problem using face relighting based on coupled bilinear model,” *Sensors*, vol. 19, no. 1, p. 43. Dec. 2018.

[4] B. Ríos-Sánchez, D. Costa-da-Silva, N. Martín-Yuste, and C. Sánchez-Ávila, “Deep learning for facial recognition on single sample per person scenarios with varied capturing conditions,” *Applied Sciences*, vol. 9, no. 24, p. 5474, 2019.

[5] D. Chen, F. Liu, and Z. Li, “Deep learning based single sample per person face recognition: A survey,” *ArXiv Preprint*, ArXiv: # 2006.11395, 2020.

[6] J. Zeng, X. Zhao, J. Gan, C. Mai, Y. Zhai, and F. Wang, “Deep convolutional neural network used in single sample per person face recognition,” *Computational Intelligence and Neuroscience*, vol. 2018, 2018.

[7] G. Mishra, P. V. Vishwakarma, and A. Aggarwal, “Constrained L1-optimal sparse representation technique for face recognition,” *Optics & Laser Technology*, vol. 129, 2020.

[8] H. Ouanan, M. Ouanan, and B. Aksasse, “Non-linear dictionary representation of deep features for face recognition from a single sample per person,” *Procedia Computer Science*, vol. 127, pp. 114-122, 2018.

[9] Y. Hui-xian and C. Yong-yong, “Adaptively weighted orthogonal gradient binary pattern for single sample face recognition under varying illumination,” *IET Biometrics*, vol. 5, no. 2, pp. 76-82, Jun. 2016.

[10] S. Allagwail, O. S. Gedik, and J. Rahebi, “Face recognition with symmetrical face training samples based on local binary patterns and the gabor filter,” *Symmetry*, vol. 11, no. 2, pp. 157, Jan. 2019.

[11] F. Liu, Y. Ding, S. Yang, and F. Xu, “Patch based semi-supervised linear regression for single sample face recognition,” in *Proc. IEEE Third Int. Conf. on Multimedia Big Data (BigMM)*, Laguna Hills, CA, 2017, pp. 62-65..

[12] T. Pei, L. Zhang, B. Wang, F. Li, and Z. Zhang, “Decision pyramid classifier for face recognition under complex variations using single sample per person,” *Pattern Recognition*, vol. 64, pp. 305-313, 2017.

[13] G. Stein, Y. Li, and Y. Wang, “One sample per person facial recognition with local binary patterns and image sub-grids,” in *Proc. Annual Conf. on Information Science and Systems (CISS)*, Princeton, NJ, 2016, pp. 7-11.

[14] M. Pang, Y. Cheung, B. Wang, and J. Lou, “Synergistic generic learning for face recognition from a contaminated single sample per person,” *IEEE Trans. on Information Forensics and Security*, vol. 15, pp. 195-209, 2020.

[15] R. Min, S. Xu, and Z. Cui, “Single-sample face recognition based on feature expansion,” *IEEE Access*, vol. 7, pp. 45219-45229, 2019.

[16] V. Cuculo, A. D'Amelio, G. Grossi, R. Lanzarotti, and J. Lin, “Robust single-sample face recognition by sparsity-driven sub-dictionary learning using deep features,” *Sensors*, vol. 19, no. 1, p. 146, Jan 2019.

[17] F. Liu, Y. Ding, F. Xu, and Q. Ye, “Learning low-rank regularized generic representation with block-sparse structure for single sample face recognition,” *IEEE Access*, vol. 7, pp. 30573-30587, 2019.

[18] N. Zhu and S. Chen, “Weighted sparse representation based on virtual test samples for face recognition,” *Optik*, vol. 140, pp. 853-859, 2017.

[19] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, “Feature transfer learning for face recognition with under-represented data,” in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR 2019)*, Long Beach, CA, 2019, pp. 5697-5706.

[20] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, “Few-shot adversarial learning of realistic neural talking head models,” in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 9458-9467.

[21] T. Wannakijmongkol, I. Khornrakhun, and T. H. Chalidabhongse, “An improved adaptive discriminant analysis for single sample face recognition,” in *Proc. 11th Int. Joint Conf. on Computer Science and Software Engineering (JCSSE)*, Chon Buri, 2014, pp. 7-11.

[22] F. Abdolali and S. Seyyedsalehi, “Improving pose manifold and virtual images using bidirectional neural networks in face recognition using single image per person,” in *Proc. Int. Symposium on Artificial Intelligence and Signal Processing (AISP)*, Tehran, 2011, pp. 37-42.

[23] C. Hu, X. Lu, M. Ye, and W. Zeng, “Singular value decomposition and local near neighbors for face recognition under varying illumination,” *Pattern Recognition*, vol. 64, pp. 60-83, Apr. 2017.

[24] B. T. Tinh, T. T. Nhan, and D. N. Chau, “A method to reduce the computational cost of Modified Hausdorff Distance in Face Recognition,” in *Proc. 6th NAFOSTED Conf. on Information and Computer Science (NICS)*, Hanoi, Vietnam, 2019, pp. 103-108.

[25] H. Tan, Y. J. Zhang, W. Wang, G. Feng, H. Xiong, J. Zhang, and Y. Li, “Edge eigenface weighted hausdorff distance for face recognition,” *Int. Journal of Computational Intelligence Systems*, vol. 4, no. 6, pp. 1422-1429, Mar. 2012.

[26] C. Dang and T. Do, “A modification of modified hausdorff distance method applying for face recognition,” *Science and Technology Development Journal*, vol. 20, no. K3, pp. 126-131, 2017.

[27] H. Tan and Y. Zhang, “Computing eigenface from edge images for face recognition based on Hausdorff distance,” in *Proc. Fourth Int. Conf. on Image and Graphics (ICIG 2007)*, Sichuan, China, 2007, pp. 639-644.

[28] C. Dang and T. Do, “A modification of line Hausdorff distance for face recognition to reduce computational cost,” *Science and Technology Development Journal*, vol. 20, no. K3, 152-158, 2017.

[29] C. H. T. Yang, S. H. Lai, and L. W. Chang, “Hybrid image matching combining Hausdorff distance with normalized gradient matching,” *Pattern Recognition*, vol. 40, no. 4, pp. 1173-1181, Apr. 2017.

[30] C. Dang-Nguyen and T. Do-Hong, “Robust line hausdorff distance for face recognition,” in *Proc. Int. Symposium on Electrical and Electronics Engineering (ISEE)*, Ho Chi Minh, Vietnam, 2019, pp. 103-107.

[31] Y. Gao, “Efficiently comparing face images using a modified Hausdorff distance,” *IEE Proceedings - Vision, Image and Signal Processing*, vol. 150, no. 6, pp. 346-350, Dec. 2003.

[32] D. Papadias, Y. Tao, K. Mouratidis, and C. K. Hui, “Aggregate nearest neighbor queries in spatial databases,” *ACM Trans. on Database Systems*, vol. 30, no. 2, pp. 529–576, Jun. 2005.

[33] S. Nutanong, E. H. Jacox, and H. Samet, “An incremental Hausdorff distance calculation algorithm,” *Proc. of the VLDB Endowment*, vol. 4, no. 8, pp. 506–517, May 2011.

[34] A. A. Taha and A. Hanbury, “An efficient algorithm for calculating the exact Hausdorff distance,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, pp. 2153-2163, Nov. 2015.

[35] D. Zhang, F. He, S. Han, L. Zou, Y. Wu, and Y. Chen, “An efficient approach to directly compute the exact Hausdorff distance for 3D point sets,” *Integrated Computer-Aided Engineering*, vol. 24, no. 3, pp. 261-277, Jan. 2017.

[36] Y. Chen, F. He, Y. Wu, and N. Hou, “A local start search algorithm to compute exact Hausdorff Distance for arbitrary point sets,” *Pattern Recognition*, vol. 67, pp. 139-148, Jul. 2017.

[37] D. Zhang, L. Zou, Y. Chen, and F. He, “Efficient and accurate Hausdorff distance computation based on diffusion search,” *IEEE Access*, vol. 6, pp. 1350-1361, 2018.

[38] H. Alt and L. J. Guibas, “Discrete geometric shapes: Matching, interpolation, and approximation,” in *Handbook of Computational Geometry*, Elsevier, 2000, pp. 121–153.

[39] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, “Comparing images using the Hausdorff distance,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 850–863, 1993.

[40] M. K. Leung and Y. H. Yang, "Dynamic two-strip algorithm in curve fitting," *Pattern Recognition*, vol. 23, no. 1, pp. 69-79, 1990.

[41] G. M. Morton, *A Computer Oriented Geodetic Data Base and a New Technique in File Sequencing*, International Business Machines Company New York, 1966.

[42] Face database of Bern University. [Online]. Available: http:// www.fki.inf.unibe.ch/databases/iam-faces-database/FullFaces.tgz

[43] A. Martinez and R. Benavente, "The AR face database," CVC Technical Report, no. 24, Jun. 1998.

Chau Dang-Nguyen received the B.E. (2010), degrees in telecommunication engineering from Ho Chi Minh City University of Technology, Vietnam and the M.E. (2013), degrees in electrical and information engineering from Toyohashi University of Technology, Japan. He is currently a Ph.D. student, Ho Chi Minh City University of Technology, Vietnam. His current interests include pattern recognition, face recognition.

Tuan Do-Hong received the B.E. (1994), M.E. (1997), degrees in telecommunication engineering from Ho Chi Minh City University of Technology, Vietnam and M.E. (2000), Ph.D. (2004) degrees in telecommunication engineering from Munich University of Technology, Germany. He is an associate professor, Department of Electrical and Electronic Engineering, Ho Chi Minh City University of Technology. His current interests include signal processing for wireless communication and digital image processing.