# A Study of Hough Transform-based Fingerprint Alignment Algorithms

Cynthia S. Mlambo

Council for Scientific

and Industrial Research

Department of Engineering,

University of Johannesburg
Pretoria, South Africa

Mmamelatelo E. Mathekga

Council for Scientific and

Industrial Research

Pretoria, South Africa

Fulufhelo V. Nelwamondo

Council for Scientific and

Industrial Research

Department of Engineering,

University of Johannesburg
Pretoria, South Africa

## ABSTRACT

This paper classify existing Hough Transform fingerprint alignment algorithms and compare their performance to determine the one that gives optimal alignment results (translation and rotation). The classification is performed by considering the implementation of each algorithm. The comparison is performed by considering the alignment results computed using each group of algorithms when varying number of minutiae points, rotation angle, and translation. In addition, the memory usage, computing time and accuracy are taken into consideration. The experiments were performed on a small database where fingerprints were captured in different orientations and locations and on the public database FVC2004. Three classes of Hough Transform-Based approaches were classified as the Local Match Based Alignment(LMBA), Discretized Rotation Based Alignment(DRBA) and Matching Pair Based Alignment (MPBA). The results revealed good accuracy on all three approaches, however, the computing time and memory usage affected the performance of each approach. The LMBA approach perform better than the DRBA and the MPBA approaches on minutiae points set with larger rotation and small number of points. The DRBA approach was found to perform better with minutiae points with large amount of translation, and the computational time was less than that of LMBA and the MPBA approaches. However, the memory usage required in DRBA and MPBA for the accumulator array is greater than memory required in LMBA.

## General Terms:

Alignment parameters, Hough Transform, minutiae points, memory usage, computing time

## Keywords:

Fingerprints, alignment, translation, rotation

## 1. INTRODUCTION

The common challenge in fingerprint recognition systems is the variations of fingerprint features extracted from two fingerprint images of the same finger captured at different instances [1]. As a result, the coordinates and orientations of corresponding fingerprint features differ. In addition, the number of features extracted from different fingerprint images is generally not equal because different regions of the fingerprint image could have been captured. In other cases, some of the features might not be extracted because of the fingerprint image quality or an error with the feature extractor. The most influential variations considered in this research are caused by transformation namely; rotation and translation. Normally, this occurs because the finger is not always placed at the same position (resulting in translation) and orientation (resulting in rotation) on the scanner. For example, a rotation or translation of 2 mm can result in a shift of approximately 40 pixels of the fingerprint image, making fingerprint recognition a challenging problem. In case of minutiae based fingerprint matching, these changes of translation and rotation can lead to corresponding minutiae from the same finger having a different location and orientation, hence:

(1) Two sets of minutiae points that are from the same finger usually have unequal number of minutiae points,

(2) Not all minutiae in each set have a corresponding point in the other set,

(3) Corresponding minutiae can be represented by different location and orientation values.

Nevertheless, one of the most presented solutions to alignment problem is to perform alignment before or during the matching process. The purpose of alignment is to simplify the process of determining corresponding features extracted from the same finger. Alignment is performed by first determining alignment parameters (rotation and translation) and then superimposing fingerprint features using determined parameters. Performing alignment simplifies the process of finding corresponding points, because the alignment involves modifying minutiae point data of one or both minutiae point sets to ensure that location and orientation values of corresponding minutiae point are approximately equal. Through this the problem of matching can be simplified into a problem of determining corresponding points from the aligned fingerprint representations.

In this study presented is the identification and classification of different ways that Hough Transform (HT) is used for the computation of the rotation angle and translation vector required to align two sets of minutiae points. Classified approaches are then

implemented, and their performances with different presentations of minutiae points extracted in different impressions of the same finger are compared. This study will contribute by presenting the classification, performance and requirements of the wide variety existing HT-based fingerprint alignment methods and for assisting in the selection of the most suitable method for translated and rotated fingerprint minutiae points. The organization and layout of this paper is as follows. Section 2 presents a background and related work on fingerprint alignment with HT approach. Section 3 present the problem formulation on this study and in Section 4 is the method on how the problem is solved. Section 5 describes classification of existing HT approaches. Section 6 describes the experimental results and discussions of the results, finally the conclusions are given in Section 7.

## 2.  LITERATURE REVIEW

The use of HT approach in fingerprint alignment started in 1990's, [2, 3, 4]. In 1996 [2] Ratha *et al* introduced HT in fingerprint matching by deriving an algorithm from the general Hough Transform (GHT) and applied it in minutiae-based fingerprint matching to determine transformation between two sets of minutiae points. An alternative approach was presented in [3] by Germain *et al* which is also derived from the GHT and improved it by first determining corresponding triangles that are formed using three minutiae points. Another alternative approach was presented by Cheng *et al* in [4] which try to reduce the accumulator array size by representing the alignment parameters by only the scale and rotation angle. Different work has been presented that improve on these early works in [2, 3, 4]. Recent work shows different improvements but still there is an unbalanced in computing time and memory usage. The balance is important because the alignment stage is a pre matching, normally the matching stage require more memory, complexity and or even more features to prove that given minutiae points are really from the same finger. Pan-Gil *et al* in [5], Pan-Moon *et al* in [6] and Lomte and Nikam in [7] presented improvement on Ratha *et al*'s [2] approach. The aim of their work was to introduce an idea of reducing required memory usage for accumulator array when embedded on systems with limited computing resources such as memory space in smart cards. This was done by implementing the alignment more accurately, and performed more computations at from a coarse-grain to fine grain resolution on the accumulator array. However, the process or iterating after computing each set of alignment parameters and performing alignment, required more computing time.

Few years after Ratha *et al* [2] presented their work, different improvements was presented. In [8] Chouta *et al* introduced a similar approach that improves on the error of transformation and deformation caused by the amount of pressure applied during the finger scanning process. The difference with the work of Ratha *et al* [2], is that its includes translation and rotation tolerances to overcome transformation and distortion errors. In addition, it does not include the checking of direction difference for each possible pair of minutiae. The aim of his work was to achieve more accurate results from the approach of Ratha *et al* [2]. The approach of Chouta *et al* [8] was further improved in 2013 by Mael *et al* [9] to reduce time and memory on accessing the accumulator array. They suggested that to improve the processing time, the construction of the accumulator array is not done on the whole template minutia set. Thus, it includes each minutia of the input set, with minutiae of the template set, that have the difference in orientation angles within a predefined threshold. Also, to sort template minutiae with regard to the minutiae orientation. With the

results of these algorithms it revealed that by using HT approaches, the accuracy of alignment can be achieved from voting for the fine nearest neighbours of the bins into the accumulator array.

In 2013 Paulino *et al* [10] applied Ratha *et al*'s [2] approach in latent fingerprint matching to present a descriptor-based Hough transform. This involved the use of orientation field and minutiae information to accumulate evidence into accumulator array *A*. For accurateness, from the selected alignment parameters further process is involved to compute more accurate results. This process consists of using the minutiae pairs that vote for a peak to compute a rigid transformation between the latent and rolled fingerprints. The results showed that this algorithm outperforms well optimized commercial also its fusion with commercial fingerprint matchers leads to improved matching accuracy. However, the memory usage was not reduced by this approach.

In 2013 Chen *et al* [11] proposed a novel hierarchical minutiae matching algorithm for fingerprint identification systems which builds upon Ratha *et al*'s [2] work. This method apply Hough Transform approach to estimate alignment parameters so that images can be registered with respect to each other. This was performed by reducing computing time when performing fingerprint alignment. With this algorithm, the alignment process is decomposed to $H$ steps, and each step uses $n/H$ minutiae of the input fingerprint ($I$) to perform the alignment with the template fingerprint in the database. Where $n$ is the total number of minutiae in $I$. The results shows that it can save much time while preserving a high identification rate, but the memory usage its still too much. In addition, the reputation of alignment process requires more computing time.

In 2013 Zhou *et al* [12] applied Hough Transform approach to improve on Scale Invariant Feature Transformation (SIFT) method used in minutiae matching. The aim of their work was to achieve a low computational resource consumption such as memory usage. This was performed by implementing a matching process that involves the Hough Transform (HT) method to perform the alignment process. This approach is similar to the approach of Ratha *et al* [2], except that the alignment was performed for all closest pairs calculated by Euclidean Distance. Also, the voting is performed on a 2-dimensional and 1-dimensional accumulator arrays for translation and rotation values, respectively. The results revealed that time for searching peaks from a 3-dimensional accumulator array is decreased, and the memory requirement can also be decreased. The results of the overall algorithm indicated that after involving HT, a significant improvement was achieved with respect to speed and accuracy in representative databases compared with the conventional minutiae-based method.

An alternative approach presented by Germain *et al* [3] in 1996 which first estimate corresponding points for example by computing matching triangles between two input and template sets. Then for each corresponding point translation and rotation values are computed, and alignment is performed to compute number of points that were able to be aligned. With this method the process of finding corresponding triangles and points also performing alignment for each computed parameters, requires too much computing time.

In 1999 Bebis *et al* [13] presented an approach for determining transformation between the input and template minutiae. This [13] approach combines the work presented by Germain *et al* [3] with the match from local evidence taken as the matching triangles formed from input minutiae set (*I*) and template minutiae set (*T*). With this approach, corresponding minutiae that form matching triangles are stored accordingly, and used to determine transformation parameters that vote in the accumulator array.

Given this information, the transformation that best maps the input triangle to the template triangle is computed. The aim of their work was to implement a registration process different impressions of the same finger. The experimental results with this alignment approach revealed that the performance of the algorithms has low computational requirements and is also fast.

In 2006 Wang and Gavrilova [14] presented an approach that combines the ideas for [2] and [13]. Instead of considering corresponding triangles they use corresponding edges of triangles. For each computed triangle they determine matching edges and corresponding minutiae to compute alignment parameters. For each computed alignment parameters they compute votes into $A$, as it is performed in the approach in Ratha *et al*'s algorithm [2]. Wang and Gavrilova involved more features to determine corresponding edges as described in [14]. The work presented by Bebis *et al* [13] has been improved to estimate more accurate matching pairs before applying HT, in [15], [16], and [17], by presenting more accurate methods of estimating corresponding pairs. On the resent work, the results revealed that if a good combination of minutiae, such as triangles can be used to estimate matching pairs, then a better alignment using HT can be achieved.

Another alternative approach was presented by Cheng *et al* [4] in 1996, where the scaling factor and rotation angle is first computed using hough transform approach, then using further stages full transformation (rotation and translation) are then computed. with this approach complex processing is requires which uses lots of memory and computing time. A motive on this approach is that transformation $(\Delta_x, \Delta_y, \theta, s)$, which aligns the two segments, must necessarily involve a scale change by an amount given by the ratio of the two segment lengths, and a rotation by an angle equal to the difference between the two segment angles. After determining a set of corresponding points and corresponding transformation (scale and angle) further stages are taken to determine optimal complete alignment parameters which includes both translation and rotation parameters. This work was further used in different matching algorithms and improved, such as in [18], [19], [20], [21] and [22].

## 3. PROBLEM FORMULATION

Having presented different implementations of the Hough Transform based fingerprint alignment algorithms, it is shown that the research area on the improvement is wide and not related as from the recent years. As from the early 2000's to the late 2013, the improvements still relate and improves on the early work presented in the 1990's, as presented in [5], [14], [8], [17], [11], [10], and [12]. As a result, the memory requirements, computing time and performance on different conditions of minutiae points has not been presented. In addition, the current studies cannot be used to predict the efficient or optimal alignment results that could be reached by using one approach or the other depending on the type of conditions that affect minutiae points. As presented in the first section, the aim of this study is to group together these implementations and improvements in order to analyse and present the performance of grouped approaches.

## 4. METHOD

The proposed approach involves the identification of different ways that the Hough Transform is used for the computation of alignment parameters required to align two sets of minutiae points, and their implementation as well as a comparison of their performances. This approach is divided into the following tasks:

(1) Identification and classification of Hough Transform based fingerprint alignment algorithms that use minutiae points.
(2) Formulation of a set of comparison measures.
(3) Implementation of identified alignment algorithms in Task 1.
(4) Comparison of identified fingerprint alignment algorithms, according to the criterion formulated in Task 2.
(5) Analyses of the results from Task 4, and drawing conclusions.

## 5. CLASSIFICATION OF HT-BASED FINGERPRINT ALIGNMENT ALGORITHMS

In this section the LMBA, DRBA and MPBA approaches are briefly explained with the emphasis of their difference with relative to the general Hough Transform approach. the classification of these approaches are derived and extended from the previous work in [23]. In general, HT-based fingerprint alignment approaches are used in different ways which can be classified into two main approaches. The first approach is summarized as a Discretized Rotation Based Alignment (DRBA). On this approach, to compute alignment parameters $AP$, each discretized rotation angle and pairs of minutiae points that give direction difference less than a defined threshold are used. In addition, each discretized rotation angle is added into each orientation of $I$. Then, for each pairing minutiae found to have allowed direction difference, add votes for all feasible translations and rotations. The second approach is summarized as a Local Match Based Alignment (LMBA). On this approach, firstly a local match between all minutiae points is determined, such as minutiae with corresponding triangles. The corresponding minutiae points are hen used to compute $AP$. Then, for each $AP$, minutiae points are aligned to determine $N$ number of aligned points and $N$ is added into accumulator array. The AP that receives the maximum number of votes is deemed to be the best transformation. In the MPBA, alignment parameters are computed from corresponding pairs of minutiae points with matching scaling factor and rotation difference.

### 5.1 DRBA Approach

The DRBA approach has been implemented in different ways, as explained in [11], [10] and [12]. In the DRBA approach, it is common to consider all given points from $I$ and $T$ as possible corresponding points. In addition, by checking if the direction difference of minutiae orientation is less than a defined threshold. The second stage is to estimate $AP$ from estimated corresponding pairs. The rotation angle is taken from the discretized data and used to compute $AP$. Translation parameters are computed using the affine transformation with the rotation angle from discretization data. At the third stage the accumulator array $A$ is required to store all possible $AP$. The bin size is used to specify the step size in $A$ and it is used when voting for the nearest bins of the current estimated $AP$. During the voting procedure it is general to cast the votes on the nearest bins, and the bin sizes are experimentally defined by considering different values from too small to large amounts. The number of votes is accumulated by adding a vote for each computed parameters.

### 5.2 LMBA Approach

The LMBA approach also has been implemented in different ways, [24], [25], [16] and [17]. The first stage in this case is performed by using some methods to determine matching points, for example: by finding pair of points with similar Euclidean distance from their locations; or by first determining corresponding triangles between

**Algorithm 1:** An Alignment Algorithm for Discretized Rotation Based Approach

---

**Input:** Sets of minutiae points from $I$ and $T$.
**Output:** Set of $(\Delta_x, \Delta_y, \Delta_\theta)$.
Set direction tolerance $\theta_0$, and initialize Hough space into 3D array
Discretize values for $(\Delta_x, \Delta_y, \Delta_\theta)$ by initializing $A[\Delta_x, \Delta_y, \Delta_\theta]$
**for** *each minutiae $m_t$ in $T$* **do**
  **for** *each minutiae $m_i$ in $I$* **do**
    **for** *each rotation angle $\theta^+$ in $\Delta_\theta$* **do**
      Compute direction difference $dd$ between $(\theta_i + \theta^+)$ and $(\theta_t)$
      $dd = \min(|(\theta_i + \theta^+) - \theta_t|, 360 - |(\theta_i + \theta^+) - \theta_t|)$
      **if** $dd < \theta_0$ **then**
        Compute translation parameters

$$\begin{bmatrix} \Delta_x^+ \\ \Delta_y^+ \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \end{bmatrix} - \begin{bmatrix} \cos\theta^+ & -\sin\theta^+ \\ \sin\theta^+ & \cos\theta^+ \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

        To increase accuracy and to overcome distortions induced during finger image acquisition, $\Delta_x^+, \Delta_y^+, \theta^+$ are quantized to the nearest bins
        **for** $\Delta_\theta = (\theta^+ - Err_\theta) \rightarrow (\theta^+ + Err_\theta)$ **do**
          **for** $\Delta_x = (\Delta_x^+ - Err_\theta) \rightarrow (\Delta_x^+ + Err_x)$ **do**
            **for**
            $\Delta_y = (\Delta_y^+ - Err_y) \rightarrow (\Delta_y^+ + Err_y)$
            **do**
              Increment $(\Delta_x, \Delta_y, \Delta_\theta)$ in the accumulator array by adding evidence into accumulator $A$
              $A[\Delta_x^+, \Delta_y^+, \theta^+] = A[\Delta_x^+, \Delta_y^+, \theta^+] + 1$

Find indexes of maximum $A$, which are the most voted alignment parameters $[\Delta_x, \Delta_y, \Delta_\theta] = max(A)$

---

$I$ and $T$; or by using similar triangles from Delaunay triangulation, and then, estimate matching points from corresponding triangles. The second stage is performed by using the affine transformation with the computed rotation angle to compute $AP$. In the third stage it is common to define different bins of the accumulator array, e.g. starting from a large size of bins to the small size of bins to find the finer results of $AP$. The number of votes is accumulated by adding a number of aligned points determined after aligning points using each set of parameters. At the end of this approach, a set of $AP$ with the highest number of votes is deemed as the one that represent the best transformation of tested sets of minutiae points.

## 5.3 MPBA Approach

Another alternative HT approach in point pattern matching and minutiae alignment was introduced by Chang *et al* [4]. In this approach transformation parameters are reduced to only rotation angle and the scaling factor $(\theta, s)$. This approach depends on matching pairs that are first detected by computing alignment parameters from each pair and computing number of aligned minutiae between *T* and *I*. The alignment parameters that gives high number of aligned points is determined through a HT-based voting process. After determining a set of corresponding points and corresponding transformation (scale and angle) further stages are

**Algorithm 2:** An Alignment Algorithm for Local Match Based Approach

---

**Input:** Sets of minutiae points from $I$ and $T$.
**Output:** Set of $(\Delta_x, \Delta_y, \Delta_\theta)$.
Define $\theta_{length}, X_{length}, Y_{length}$ and number of matching levels e.g. $levels = 3$
Define required thresholds, $l_0$ for lengths of the triangles, and $a_0$ for the largest angle. Define possible alignment parameters for $(\Delta_x, \Delta_y, \Delta_\theta)$
Determine Delaunay triangles $DT_I$ and $DT_T$ for $I$ and $T$, respectively, and compute $invar\_features[]$ as discussed in Algorithm **??**
Compute alignment parameters as follows.
**if** $levels\,! = 0$ *and* $alignment\_score < set\_tolarance$ **then**
  Set the unit size for $A[\Delta_x, \Delta_y, \Delta_\theta]$
    $\Delta_{\theta_{size}} = \theta_{length}/2^{levels-1}$
    $\Delta_{x_{size}} = X_{length}/2^{levels-1}$
    $\Delta_{y_{size}} = Y_{length}/2^{levels-1}$
  Initialize $A[\Delta_{x_{size}}, \Delta_{y_{size}}, \Delta_{\theta_{size}}]$
  Set the bin size of rotation and translation
    $\Delta_{binsize} = 2^{levels-1}$
  Set $(\theta_{length}, X_{length}, Y_{length}) = (\Delta_{x_{size}}, \Delta_{y_{size}}, \Delta_{\theta_{size}})$
  **for** *each triangle in $DT_T$* **do**
    **for** *each triangle in $DT_I$* **do**
      Determine a triangle in $DT_I$ that correspond with a current triangle from $DT_T$ by For each corresponding triangle, compute alignment parameters using Algorithm 3

Find index of maximum $A$ which is the most voted alignment parameters
$[\Delta_x, \Delta_y, \Delta_\theta] = max(A)$

---

**Algorithm 3:** Voting Procedure for Local Match Based Approach

---

**Input:** An accumulator array $A$.
    Two corresponding triangles from $DT_I$ and $DT_T$.
**Output:** An updated accumulator array.
**for** *each minutiae $m_t$ in $DT_T$* **do**
  **for** *each minutiae $m_i$ in $DT_I$* **do**
    Compute direction difference $\theta^+$ between $(\theta_i)$ and $(\theta_t)$
    $\theta^+ = \min(|\theta_i - \theta_t|, 360 - |\theta_i - \theta_t|)$
    Compute translation parameters

$$\begin{bmatrix} \Delta_x^+ \\ \Delta_y^+ \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \end{bmatrix} - \begin{bmatrix} \cos\theta^+ & -\sin\theta^+ \\ \sin\theta^+ & \cos\theta^+ \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

    Align $I$ and $T$ using $(\Delta_x^+, \Delta_y^+, \theta^+)$
    Calculate number of aligned points $N$.
    Add evidence $N$ into accumulator $A$ or record computed parameters and the number of matched points
    $A[\Delta_x^+, \Delta_y^+, \theta^+] = A[\Delta_x^+, \Delta_y^+, \theta^+] + N$

---

taken to determine optimal complete alignment parameters which includes both translation and rotation parameters. This work was further used in different matching algorithms and improved, such as in [18], [19], [20], [21] and [22].

## 6. EXPERIMENTAL RESULTS

The experiments were performed using a public FVC2004 in DB3B, Db4a and DB4B databases [26] to investigate the accuracy of alignment parameters as computed using the LMBA, MPBA and DRBA. The database used was developed using low cost capacitive fingerprint scanners. The database contains a total of about 880 fingerprints of 110 different fingers, with each finger represented in 8 different impressions. From each fingerprint image, minutiae points were extracted using a manual feature extraction program. Minutiae points were extracted from each finger, the first impression was used as a template $T$ and other 7 impressions as the input $I$. For each set of minutiae points, LMBA, MPBA and DRBA were used to compute alignment parameters, where the inputs to the algorithm are $I$ and $T$, the output is a set of alignment parameters $(\Delta_x, \Delta_y, \Delta_\theta)$. After computing alignment parameters, alignment was performed between fingerprint images of $I$ and $T$. The LMBA, MPBA and DRBA algorithms were implemented in Matlab and compared in terms of accuracy of alignment results, computing time and memory usage. Code implementation was performed on the computer with a processor of *Intel(R) Core(TM) i7-3770 CPU@3.40GHz.*

In addition, the performance for code implementation were evaluated by considering different impressions of fingerprint images that represent common challenges in alignment, such as unequal number of minutiae points with translation and rotation of minutiae points. The values of alignment parameters were compared with the results of a program that computes alignment parameters manually from given corresponding minutiae points.

The correctness of alignment results and the conclusions were derived from studying the results on two possible alignment errors. Alignment parameters are deemed as correct if the difference between the results of a mentioned program and each approach is less than a thresholds of 8 which was selected experimentally. The first error occurs when corresponding points are not aligned by the algorithms. A second error is generated when the results of the algorithm shows missed aligned points. Misalignment can be visually identified as points that are not corresponding but were aligned successfully.

### 6.1 Alignment Accuracy

To test the accuracy of alignment results, number of aligned minutiae points and alignment parameters were evaluated from experimental results. The individual bar graphs from Figure 1 to Figure **??** show results on the accuracy of the LMBA, MPBA and DRBA approaches in different conditions that affect minutiae points.

*6.1.1 Rotation.* The performance of the DRBA, the MPBA and the LMBA on the amount of rotation is summarised in Figure 1. The results indicate that the correctness of alignment for both approaches decreases as the rotation amount increases. However, the LMBA performs better than the DRBA and MPBA even though its performance also decreases. This is because LMBA and DRBA approaches involve certain limitations on the rotation angle. With the DRBA, only minutiae point pairs with the direction difference less than a predefined threshold are allowed to cast votes in $A$. As a result, some corresponding minutiae points are treated as not corresponding if their orientation angle is faulty, for example due to the feature extractor. In the LMBA approach, a difference of the largest angles of corresponding triangles should be less than a predefined threshold. This results in corresponding points being treated as not matching if there are missing and additional points
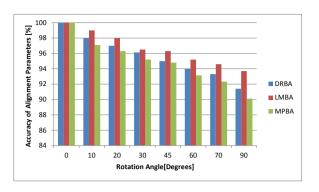


Fig. 1: Performance comparison with the amount of rotation.

in $I$, where new triangles are formed with different internal angles. With these limitations, corresponding minutiae points with a large difference of rotation values were not aligned properly from the alignment results. However, the rotation values are mostly correct compared to the translation values, because during the rotation it is common that minutiae points are skewed. In addition, the amount of translation for translated points in the same set is not linear.

*6.1.2 Translation.* When the position of the finger changes different portions of fingerprints of the same finger are captured. As a result, not all minutiae points captured at different instances get captured on another instance. However, if there is a portion if fingerprint image similar with the other fingerprint captured there are corresponding points between two fingerprint images. While if different portions were captured, all used algorithms found that there is no alignment between two given images. With the DRBA, as it uses all pair combinations of minutiae points, the alignment parameters are computed on corresponding points. In addition, if there is only translation without rotation, the DRBA preformed better than the LMBA and the MPBA, as shown in Figure 2. The LMBA and the MPBA involves the combination of two or three minutiae points with the distance between them. The performance on these approaches drops as the translation increases because additional minutiae points captured at different instances can have similar distance compared to missing points. as a result, misalignment depended in length of minutiae points occurs.

*6.1.3 Number of Minutiae Points.* The experimental results indicate that as the translation and rotation increases, some minutiae points are removed and new minutiae from different portions of fingerprints are added. As a result, Figure 3 illustrates the correctness of alignment parameters determined as the number of minutiae points increases. The results shows that the LMBA is able to perform better even if the number of points is small compared to the DRBA. However, in both approaches the correctness of alignment parameters increases as the number of points increase. This is because with the LMBA approach alignment parameters are computed from corresponding triangles, therefore if there are efficient corresponding triangles between input and template sets, enough evidence can be accumulated. Where else with the DRBA, each pair of minutiae points from the input and template add a vote into the accumulator array,

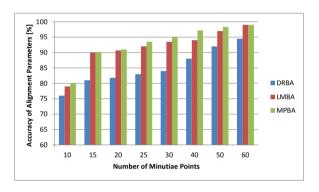Fig. 2: Performance comparison with the amount of translation.



Fig. 3: Performance comparison with the number of minutiae points.

which leads to the accumulation of few evidence. In the MPBA, since it depends on matching points according to the distance and the angle formed by two minutiae points, if there are enough corresponding points that gives constant rotation and translation between corresponding points, the alignment results becomes true.

## 6.2  Computing Time

Rotation and translation affect the number of points, therefore time was measured by analysing time taken when there are missing or additional minutiae points caused by the rotation or translation. The experiments were run on the computer with Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz and installed memory (RAM) of 8GB. Time was measured using Matlab function for measuring computing time. An average of time was then calculated from the time measured on rotated and translated fingerprint images with different number of minutiae points.

The average time taken to compute alignment parameters during the ideal and realistic scenario experiments was measured to have an average of 3.84s, 3.12s and 0.40s for LMBA, MPBA and DRBA, respectively. The results showed that small amounts of
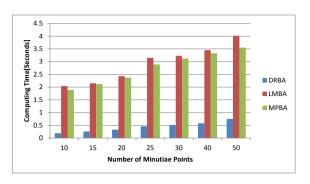


Fig. 4: Average computing time.

rotation has no effect on the processing time. However, when the rotation increases the number of minutiae points changes. If the number of minutiae decreases, the amount of time becomes less than that required when the image was not rotated, as presented in Figure 4. On the other hand when the number of minutiae increases, the required computing time becomes greater. In addition, when there is translation, the experimental results showed that for both the LMBA, MPBA and DRBA time taken to compute alignment parameters decreases as translation decreases because the number of minutiae points decreases.

## 6.3  Memory Usage

To determine required memory for each approach, memory measurements were performed on the variables used in code implementation. In addition, since rotation and translation affect the number of points, memory usage was measured when analysing the results from missing or additional number of points caused by the rotation or translation.

The most effective variable in both LMBA, DRBA and MPBA is the accumulator array $A$. The declaration and the initialization of the accumulator array is generally challenging because when it is defined with large size, the results are not accurate. This is caused by the large amount of difference between the true alignment results and the computed ones. For example, if the bin size is 20, an alignment parameters of 7 will be deemed as 20, which makes a big difference. A bigger bin size means there are fewer bins in $A$. In addition, the memory allocation required for the accumulator array becomes efficient for small memory applications (e.g. smart cards). On the other hand, if the bin size is too small, the accuracy of alignment parameters is good. However, the memory requirements increases and makes it challenging for use on small memory applications.

The results are summarized in Figure 5. The results represent that DRBA, LMBA and MPBA are affected by the way $A$ is defined. Normally, the accumulator array in the LMBA algorithms is firstly defined with larger bin size because this method involves several stages where the bin sizes are reduced. Therefore, even if the bin size is approximately 20 as shown in Figure 5, the results will be accurate after the bin size is reduced. While with the DRBA it performs better with the bin size close to 10. With the MPBA it gives accurate results when the bin size is less than 10, and as the bin size increases the accuracy on alignment parameters drops.
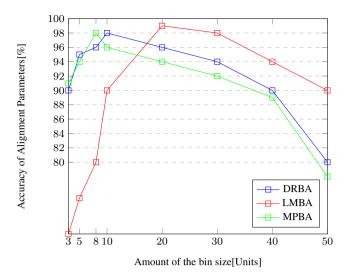
Fig. 5: Graph results for both DRBA, MPBA and LMBA approaches

Table 1. : Average performance in memory, computing time and accuracy.

| Metric | NumberOfPoints | Accuracy | Time | Memory |
|---|---|---|---|---|
| NumberOfPoints | DRBA | 93% | 0.41s | 1.06KB |
| | MPBA | 99% | 3.55s | 0.84KB |
| | LMBA | 98% | 4.67s | 0.13KB |
| Translation | DRBA | 90% | 0.33s | 1.06KB |
| | MPBA | 88% | 3.37s | 0.84KB |
| | LMBA | 85% | 4.71s | 0.13KB |
| Rotation | DRBA | 80% | 0.39s | 1.06KB |
| | MPBA | 77% | 3.86s | 0.84KB |
| | LMBA | 85% | 4.52s | 0.13KB |

The LMBA and DRBA require certain thresholds to test if computed parameters for minutiae points are accepted, as detailed in Algorithm 1 and Algorithm 2. In this experiment, the effect of the size of thresholds was tested. The results showed that with the DRBA, which required threshold for a direction difference, the accuracy of alignment parameters changes as the size of direction difference changes. In addition, the results are affected when the rotation is higher than a predefined threshold. In the experiments for the realistic scenario, this threshold was kept to an average size of 60, because the normal rotation of the fingerprint is less than 30. The LMBA requires a threshold for the ratios of the largest angles in each triangle in $I$ and $T$, and for difference of the ratios formed from the two smallest segments with largest length in each triangle. The results showed that accurate results are achieved when the thresholds are set greater or equal to 0.5. This is because if the thresholds are large, enough combination of points will be allowed to compute alignment parameters and add evidence into $A$. As a result LMBA requires more memory compared to MPBA and DRBA. The results shows that when the number of minutiae points increases , in the LMBA the number of triangles increases. As a result it uses more memory to compute triangles and to determine corresponding triangles. On the other hand, in the DRBA, only computing time is affected when number of minutiae points changes, the memory is kept constant. In addition, in both approaches memory allocation for $I$ and $T$ changes as the number of minutiae points changes. During the experiments, a small increase of memory on the DRBA was caused by $I$ and $T$ variables.

### 6.4 Summary on Comparison

Table 1 represent the summary of the results for the experiments performed in the selected images of FVC2004 database with the performance evaluation of each approach.
The memory usage is the measured as from required to accumulate evidences in the accumulator array $A$. In light of these results, it is clear that the DRBA can be used in any situation of fingerprints that affected by the translation and rotation, especially on high amount of rotation and few number of minutiae points. The LMBA can also be used where there is translation and rotation but with

the expectancy that when the rotation and translation amount is greater the used thresholds the results will not be accurate. In the MPBA, the accuracy is very good when there is no rotation but due to that rotation causes missing points and the length between two points changes, it cannot be suitable. In addition the memory space required in the DRBA is greater than the one required in the LMBA and MPBA. This is because the bin sizes in the accumulator array of the DRBA approach is less than that of the LMBA approach. However, the computing time required in the MPBA approach is greater than that required in the LMBA and much less that that in the DRBA. This is because the implementation of the LMBA performs alignment in each set of computed alignment parameters, and its procedure is performed in several levels. In addition, the implementation of the MPBA involves complex computation that requires iterations which takes longer time to compute alignment results.

### 7. CONCLUSION

The classification and comparison between three main HT-based fingerprint alignment algorithms is presented. These algorithms involves the process of adding a vote into the accumulator array, which leads to the accumulation of few evidence. HT based fingerprint alignment approaches presented, are Discretized Orientation Based Approach (DRBA), Matching Pair Based Alignment (MPBA) and Local Match Based Approach (LMBA). The AP is computed from a FVC2004 database which contains different rotated and translated fingerprints impressions of the same finger. The experimental results revealed good accuracy on all three approaches, however, the computing time and memory usage affected the performance of each approach. The LMBA approach perform better than the DRBA and the MPBA approaches on minutiae points set with larger rotation and small number of points. The DRBA approach was found to perform better with minutiae points with large amount of translation, and the computational time was less than that of LMBA and the MPBA approaches. However, the memory usage required in DRBA and MPBA for the accumulator array is greater than memory required in LMBA. In addition, the future work will involve the study of the effects that are caused by the limitations in each approach. For instance, improving on the efficient method of defining the accumulator array that requires less memory and computing time.

### 8. REFERENCES

[1] J. Vacca, *Biometric Technologies and Verification Systems.* Butterworth-Heinemann, 2007.

[2] N. K. Ratha, K. Karu, S. Chen, and A. Jain, "A real-time matching system for large fingerprint databases," *IEEE*

*Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 799–813, 1996.

[3] R. Germain, A. Califano, and S. Colville, "Fingerprint matching using transformation parameter clustering," *Journal of the Computational Science and Engineering, IEEE*, vol. 4, no. 4, pp. 42–49, 1997.

[4] S.-H. Chang, F.-H. Cheng, W.-H. Hsu, and G.-Z. Wu, "Fast algorithm for point pattern matching: invariant to translations, rotations and scale changes," *Journal of Pattern Recognition*, vol. 30, no. 2, p. 311320, 1997.

[5] S. Pan, Y. Gil, D. Moon, Y. Chung, and C. Park, "A memory efficient fingerprint verification algorithm using a multi resolution accumulator array," *Journal of the Electronics and Telecommunications Research Institute (ETRI)*, vol. 25, no. 3, pp. 179–186, 2003.

[6] S. Pan, D. Moon, and K. Kim, "A fingerprint matching hardware for smart cards," *Journal of the Institute of Electronics, Information and Communication Engineers (IEICE) Electronics Express*, vol. 5, no. 4, pp. 136–144, 2009.

[7] A. Lomte and S. Nikam, "Biometric fingerprint authentication by minutiae extraction using USB token system," *International Journal Computer Technology and Applications*, vol. 4, no. 2, pp. 187–191, 2013.

[8] T. Chouta, J. Danger, L. Sauvage, and T. Graba, "A small and high-performance coprocessor for fingerprint match-on-card," pp. 915–922, 15th Euromicro Conference on Digital System Design, IEEE, 2012.

[9] B. Mael, Y. Bocktaels, J. Bringer, H. Chabanne, T. Chouta, J. Danger, M. Favre, and T. Graba, "Studying potential side channel leakages on an embedded biometric comparison system," in *Constructive Side-Channel Analysis and Secure Design*, pp. 281–298, Springer International Publishing, 2014.

[10] A. Paulino, J. Feng, and A. Jain, "Latent fingerprint matching using descriptor-based Hough transform," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 31–45, 2013.

[11] F. Chen, X. Huang, and J. Zhou, "Hierarchical minutiae matching for fingerprint and palm print identification," *IEEE Transactions on Image Processing*, vol. 22, no. 11-12, pp. 4964–4971, 2013.

[12] R. Zhou, D. Zhong, and J. Han, "Fingerprint identification using SIFT-based minutia descriptors and improved all descriptor-pair matching," *Academic Journal of Sensors (14248220)*, vol. 13, no. 3, pp. 3142–3156, 2013.

[13] G. Bebis, T. Deaconu, and M. Georgiopoulos, "Fingerprint identification using delaunay triangulation," pp. 452–459, IEEE International Conference on Intelligence, Information and Systems (ICII'99), 1999.

[14] C. Wang and M. L. Gavrilova, "Delaunay triangulation algorithm for fingerprint matching," pp. 208–216, Proceedings of the 3rd international Sympsium on Voronoi Diagrams in Science and Engineering, IEEE, 2006.

[15] T. Uz, G. Bebis, A. Erol, and S. Prabhakar, "Minutiae-based template synthesis and matching for fingerprint authentication," *Journal of Computer Vision and Image Understanding*, vol. 113, no. 9, pp. 979–992, 2009.

[16] P. Junior, A. de Nazare-Junior, and D. Menotti, "A complete system for fingerprint authentication using Delaunay triangulation," pp. 1–7, Re-conhecimento de Padroes, DECOM UFOP, 2010.

[17] A. Gheibi and A. Mohades, "Stable geometric fingerprint matching," 2013. Last accessed on 01/08/2014.

[18] V. Wamelen, Z. Li, and S. Iyengar, "A fast algorithm for the point pattern matching problem," *Transactions on IEEE, Pattern Analysis and Machine Intelligence(PAMI)*, vol. 37, 2000.

[19] V. Wamelen, B. Paul, Z. Li, and S. Iyengar, "A fast expected time algorithm for the 2-D point pattern matching problem," *Journal of Pattern Recognition*, vol. 37, no. 8, pp. 1699–1711, 2004.

[20] Y. Tong, H. Wang, D. Pi, and Q. Zhang, "Fast algorithm of Hough transform-based approaches for fingerprint matching," pp. 10425–10429, In Intelligent Control and Automation, WCICA 2006, The Sixth World Congress on IEEE, 2006.

[21] R. Singh, U. Shah, V. Gupta, and S. Dube, "Fingerprint recognition CS676,image processing and computer vision," 2009. Last accessed on 15/08/14.

[22] N. Kumar and P. Verma, "Fingerprint image enhancement and minutia matching," *International Journal of Engineering Sciences and Emerging Technologies*, vol. 2, no. 2, pp. 37–42, 2012.

[23] C. Mlambo, F. Nelwamondo, and M. Mathekga, "Comparison of effective Hough transform-based fingerprint alignment approaches," pp. 84–89, Conference in International Symposium on Biometrics and Security Technologies, IEEE, 2014.

[24] Y. Liu, D. Li, T. Isshiki, and H. Kunieda, "A novel similarity measurement for minutiae-based fingerprint verification," pp. 1–6, 2010 4th IEEE International Conference in Biometrics: Theory Applications and Systems (BTAS), 2010.

[25] D. Marius and B. Monica, "Multimodal access control systems which combines classical access control methods with biometric methods," *Journal of the 9th International Symposium on Electronics and Telecomunications (ISETC), Timisoara*, vol. 51, no. 4, pp. 17–22, 2010.

[26] FVC2000., "Fingerprint verification compatition," 2014. Last accessed on 20/07/14.