A Two-Stage Point Pattern Matching Algorithm Using Ellipse Fitting and Dual Hilbert Scans

Li TIAN⁠(a), Student Member and Sei-ichiro KAMATA⁠(b), Member

SUMMARY Point Pattern Matching (PPM) is an essential problem in many image analysis and computer vision tasks. This paper presents a two-stage algorithm for PPM problem using ellipse fitting and dual Hilbert scans. In the first matching stage, transformation parameters are coarsely estimated by using four node points of ellipses which are fitted by Weighted Least Square Fitting (WLSF). Then, Hilbert scans are used in two aspects of the second matching stage: it is applied to the similarity measure and it is also used for search space reduction. The similarity measure named Hilbert Scanning Distance (HSD) can be computed fast by converting the 2-D coordinates of 2-D points into 1-D space information using Hilbert scan. On the other hand, the N-D search space can be converted to a 1-D search space sequence by N-D Hilbert Scan and an efficient search strategy is proposed on the 1-D search space sequence. In the experiments, we use both simulated point set data and real fingerprint images to evaluate the performance of our algorithm, and our algorithm gives satisfying results both in accuracy and efficiency.

key words: point pattern matching, ellipse fitting, Hilbert scan, transformation parameter estimation, search space reduction

1. Introduction

Point Pattern Matching (PPM) is an important problem found in a variety of applications such as image registration, object recognition and stereo vision. PPM algorithm always matches two point patterns by finding an optimal transformation where a similarity (dissimilarity) measure of the two point patterns is minimized (maximized). PPM can be divided into complete and incomplete cases based on if there exists a one to one mapping between the two point sets to be matched, and it can also be divided into labelled and unlabelled cases based on if additional information such as intensity, color and gradient other than 2-D image coordinates are used [1]. Hence, PPM problems can be classified into four different types as shown in Fig. 1. Of all these four types, type 4 is the most difficult one and also is the most suitable one for describing many real-world image and vision applications. In this study, we concentrate on this type under the affine transformation. Two factors are important in this type of PPM problem: the similarity measure and the search strategy.

Many approaches have been developed for the two factors mentioned above. Two well-known similarity measures called Hausdorff distance (HD) and partial Hausdorff distance (PHD) have been widely used for PPM [2]. Modified Hausdorff distance (MHD) [3] is an extension of them. A heuristic dissimilarity measure is also proposed in [4]. On the other hand, many search strategies including clustering methods [5], [6], parameter decomposition methods [7], [8], relaxation methods [9], [10], bounded alignment [11], spectral graph analysis [12], genetic algorithms [13], simulated annealing [14] and particle swarm optimization [1] also have been proposed for PPM problems. Among them, only a few of them are suitable for type 4 and many of them have the assumption that two point patterns are already coarsely aligned, which is not feasible in real-world applications.

In this study, we propose a two-stage point pattern matching algorithm using ellipse fitting and Dual Hilbert scans. In the first (or coarse) matching stage, a four-point estimation method based on axis node points of ellipses obtained by a fitting method named Weighted Least Square Fitting (WLSF) is proposed for PPM. In the second (or fine) matching stage, Hilbert scans are used twice in the matching process: first it is applied to the similarity measure named Hilbert Scanning Distance (HSD) which can be computed fast by converting the 2-D coordinates of 2-D points into 1-D space information using Hilbert scan; and then it is used to convert the N-D search space to a 1-D search space sequence and a fast search strategy, which is similar to gradient descent, is proposed on the 1-D search space sequence. In the experiments, we evaluate our algorithm using both simulated data and real fingerprint images for PPM problem. In the first matching stage, WLSF shows more robustness to noise and outlier points than Direct Least Square Fitting (DLSF) in ellipse fitting and the search space can be reduced to a small range. The whole algorithm also gives satisfying results both in accuracy and efficiency for PPM problem.

This study gives four major contributions to ellipse fit-
ting method and PPM problem:
1. WLSF provides a more robust fitting method for noise and outlier points than DLSF in PPM problem.
2. The coarse transformation parameter estimation using four axis node points of ellipses is stable and can reduce the search space into a small range to save searching time much in PPM problem.
3. A search strategy based on 1-D Hilbert scanned search space is proposed and it has flexibility between accuracy and efficiency.
4. Making use of dual Hilbert scans for PPM is effective and it is suitable for hardware.

The remainder of this paper is organized as follows. First, we discuss our previous work including 3-D Hilbert scan and HSD in Sect. 2; then, details about the proposed WLSF and the first matching stage is given in Sect. 3; Section 4 is about the second matching stage using dual Hilbert scans; Section 5 presents experimental results using both simulated data and real fingerprint images and gives some discussions. Conclusions and future work are given in the last section.

2. Previous Work

In this section, we will introduce two important techniques related to the proposed algorithm: 3-D Hilbert Scan and HSD. Hilbert Scan is based on the Hilbert space-filling curves. Space-filling curves are curves whose ranges contain the entire 2-dimensional unit square (or the 3-dimensional unit cube, or n-dimensional hypercube). They can convert N-D space to 1-D space by one-to-one mapping.

In 1890, Peano discovered a densely self-intersecting curve named Hilbert curve which passed through every point of the unit square which is different from Lebesgue curve and Zig-zag curve. Hilbert curve has been widely applied in image processing tasks such as image compression, image clustering, pattern recognition [15]-[18] since it has the property that it preserves point neighborhoods as much as possible. Our matching algorithm is inspired and based on the mentioned work.

2.1 3-D Hilbert Scan

The Hilbert curve is a locus of points in N-D space. For its special attributes, it becomes an analytical solution of a space-filling curve. In 1891, Hilbert made a curve having the space-filling property in 2-D space. The generation of a 3-D Hilbert curve is described in [19]. It demonstrated that the sub-cubes can be arranged so that the inclusion relationships are preserved, that is, if a cube corresponds to an interval, then its sub-cubes correspond to subintervals of that interval.

A one-to-one mapping whose domain is a set of P non-negative integers (I) and whose image is a set of P points (Q), is called a scan. If a point \( a \in A \) and \( b \in B \) has integer coordinates in the 2-D space. The first matching stage is given in Sect. 3; Section 4 is about the second matching stage using dual Hilbert scans; Section 5 presents experimental results using both simulated data and real fingerprint images and gives some discussions. Conclusions and future work are given in the last section.

2.2 Hilbert Scanning Distance

HSD is proposed as a similarity measure and utilized for matching point sets. We don’t discuss the details that how to compute HSD [21] in this study. Here, we only summarize the definition and brief computing process of HSD.

Assume that we are given two finite point sets \( A = \{a_1, \ldots, a_I\} \) and \( B = \{b_1, \ldots, b_J\} \) such that each point \( a \in A \) and \( b \in B \) has integer coordinates in the 2-D space. We firstly use Hilbert scanning to convert them to new sets \( S = \{s_1, \ldots, s_I\} \) and \( T = \{t_1, \ldots, t_J\} \) in the 1-D sequence, respectively. Then, the directed HSD from \( A \) to \( B \) is computed by

\[
h_{\text{hSD}}(A, B) = \frac{1}{I} \sum_{i=1}^{I} \rho(\min_j |s_i - t_j|)
\]
where $\| \cdot \|$ is the Euclidean norm distance in the 1-D space and function $\rho$ is defined as:

$$\rho(x) = \begin{cases} x & (x \leq \tau) \\ \tau & (x > \tau) \end{cases}$$  \hfill (2)

where $\rho$ is called the threshold elimination function and $\tau$ is a threshold predefined. We also can obtain the directed HSD from $B$ to $A$ $h_{hsd}(B,A)$ similarly and HSD is defined by

$$H_{hsd}(A,B) = \max(h_{hsd}(A,B), h_{hsd}(B,A))$$  \hfill (3)

If two binary images $A$ and $B$ including feature points have been given, we first construct a new image $C$ by combining them under a certain transformation. Note that there must be 3 types of feature point in the combined image $C$: feature point from $A$, feature point from $B$ and overlapped feature point from both $A$ and $B$. In order to distinguish the 3 types of point, we give values 1, 2 and 3 to them, respectively. Then, the process of computing $h_{hsd}(A,B)$ can be summarized as the following steps:

**Step1** Using a prepared lookup table to convert the combined image to a 1-D sequence $C'$;

**Step2** For any feature point $c'_i$ in $C'$: if (value of $c'_i$) = 2, next feature point; else if (value of $c'_i$) = 1, find the nearest feature point $c''_i$ whose value is 2 or 3, and the distance $d_i = \| c'_i - c''_i \|$; else (value of $c'_i$) = 3, $d_i = 0$;

**Step3** If $d_i > \tau$, then $d_i = \tau$. This step can also be done by specifying the search range in Step 2 more efficiently;

**Step4** Repeat Step2 and Step3 until all the feature points have been processed;

**Step5** Obtain $h_{hsd}(A,B)$ by calculating the mean value of all $d_i$;

**Step6** Stop.

### 3. First Matching Stage

Fitting geometrical models to image data is important in image processing and computer vision fields since it can reduce and simplify the data for high level image processing task. Among various geometrical models, ellipse is the most commonly used one. Ellipse fitting methods can be categorized into two types: clustering methods such as Hough-based methods [22]–[25] and least square fitting methods. In this study, we focus on the latter type. Most of the least square fitting methods concentrate on finding the parameters to minimize some distance measure between the ellipse and the data points. Different constraints [26]–[30] applied to the parameters are used to obtain the solution. Many refinement methods are also developed to improve the accuracy of fitting results [31]–[33]. Among all these methods, DLSF is reported as the most robust one to noise [30]. This section introduces a modified fitting method WLSF and discuss how to estimate the transformation parameters using fitted ellipses in the first matching stage.

#### 3.1 Ellipse Fitting

A general quadratic curve can be presented as an implicit second order polynomial:

$$F(a;x) = ax^2 + bxy + cy^2 + dx + fy + g = 0$$  \hfill (4)

where $a = [a b c d f g]^T$ and $x = [x^2 \ xy \ y^2 \ x \ y \ 1]^T$. In DLSF [30], $F(a;x_i)$ is called the algebraic distance of a point $(x,y)$ to the quadratic curve $F(a;x) = 0$. In our method, we give a weight $w_i$ to each algebraic distance of a point $(x,y)$ and called it as weighted algebraic distance:

$$G(a;x_i) = w_i F(a;x_i).$$  \hfill (5)

We will discuss how to compute the weight later. The fitting of the quadratic curve can be obtained by minimizing the sum of squared weighted algebraic distances

$$Q_1(a) = \sum_{i=1}^{N} G(x_i)^2$$  \hfill (6)

of $N$ data points $x_i$. The minimization can be solved by using deficient generalized eigenvalue system as follows:

$$Q^T a = \lambda C a$$  \hfill (7)

where $Q = [x_1 x_2 \cdots x_n]$ is the design matrix and $C$ is the constraint. We use the same constraint $4ac - b^2 = 1$ as in DLSF. Now, the constrained ellipse problem can be solved by minimizing the Eq. (6) subject to this constraint. For more details about how to solve the system, please refer to [26], [30].

Now, we are beginning to discuss how to compute the weight for each point in a point set. Given a point set $P = \{ p_i = (x_i, y_i) | i = 1, 2, \ldots, n \}$ in $\mathbb{R}^2$. First, we compute the central point $p_c = (x_c, y_c)$ as

$$x_c = \frac{1}{N} \sum_{i=1}^{N} x_i, \quad y_c = \frac{1}{N} \sum_{i=1}^{N} y_i.$$  \hfill (8)

Then, for each point in this point set, compute the Euclidean distance $d_i$ from it to the central point. Finally, the weight of a point in the point set is computed as

$$w_i = \begin{cases} 1 & \text{if } d_i \leq r_1 \\ 1 - (d_i - r_1)/(r_2 - r_1) & \text{if } r_1 < d_i < r_2 \\ 0 & \text{if } d_i \geq r_2 \end{cases}$$  \hfill (9)

where $r_1$ and $r_2$ ($r_1 < r_2$) are two thresholds predefined.

Figure 3 gives an illustration that how to compute the weights: if a point is close to the central point as in region $R_1$, the weight is 1, that means it gives a full algebraic distance to the Eq. (6); if a point is a little far from the central point as in region $R_2$, its contribution to Eq. (6) is weakened

\footnote{In some matching applications, feature point also can be used instead of the central point, i.e., core point are used as the central point in fingerprint images.}
based on its distance to the central point; if a point is far away from the central point, it will be eliminated. The proposed WLSF method is similar to DLSF; particularly, if the parameter $r_1$ is large enough, WLSF becomes the same form as DLSF. However, the weights used in WLSF can help the fitting method to improve its robustness to noise and outlier points in many real matching problems such as fingerprint matching.

3.2 Parameter Estimation

In PPM problem, if the transformation space is large, it will take a long time to find the optimal transformation position. Hence, techniques for reducing the search space to a local space around the optimal transformation position are highly desired. In our study, we propose a simple but powerful method to estimate the transformation parameters using the ellipses obtained by WLSF.

Generally, linear conformal, affine, and projective transformation can be obtained by giving a pair of two, three, and four corresponding points, respectively [34]. Here, we discuss the four-point case and the two- and three-point cases can be included in the four-point case. Given a fitted ellipse, four node points of major and minor axes of the ellipse are selected to compute the projective transformation as shown in Fig. 4. Notice that there are four corresponding candidates (two in linear conformal and affine transformation): $(A \leftrightarrow A', B \leftrightarrow B', C \leftrightarrow C', D \leftrightarrow D')$, $(A \leftrightarrow C', B \leftrightarrow D', C \leftrightarrow A', D \leftrightarrow B')$, $(A \leftrightarrow A', B \leftrightarrow D', C \leftrightarrow C', D \leftrightarrow B')$, and $(A \leftrightarrow C', B \leftrightarrow B', C \leftrightarrow A', D \leftrightarrow D')$. The false three can be easily rejected by comparing some similarity measure such as correlation of the image or the Hausdorff distance [2].

The projective transformation $x' = Hx$ can be expressed in a matrix form

$$
\begin{pmatrix}
  x'_1 \\
  x'_2 \\
  x'_3
\end{pmatrix} =
\begin{bmatrix}
  h_{11} & h_{12} & h_{13} \\
  h_{21} & h_{22} & h_{23} \\
  h_{31} & h_{32} & h_{33}
\end{bmatrix}
\begin{pmatrix}
  x_1 \\
  x_2 \\
  x_3
\end{pmatrix}
$$

(10)

where $H$ is called the homogeneous matrix and $(x_1, x_2) \leftrightarrow (x'_1, x'_2)$ is the point correspondence in Cartesian coordinates. Following the work in [35], we can generate two linear equations for the elements of $H$

$$
x'_1(h_{31}x_1 + h_{32}x_2 + h_{33}) = h_{11}x_1 + h_{12}x_2 + h_{13}
$$

$$
x'_2(h_{31}x_1 + h_{32}x_2 + h_{33}) = h_{21}x_1 + h_{22}x_2 + h_{23}.
$$

(11)

If we have $n \geq 4$ points (no three points collinear), then the homogeneous matrix $H$ can be determined uniquely. After obtaining the homogeneous matrix, the transformation parameter can be easily obtained and a local search space around it can be used to search the optimal transformation in the next stage.

4. Second Matching Stage

In this section, we will explain how to obtain the optimal transformation parameters by using Hilbert scanning technique. Here, we take the 3-D Hilbert scan as the example. So, horizontal translation $\Delta x$, vertical translation $\Delta y$ and rotation $\theta$ are considered as the transformations. If the number of transformation parameters is larger than 3, it can also be solved by using N-D Hilbert scan.

Thus, the second matching stage can be listed as following steps:

1. Construct the 3-D parameter space coordinates $(\Delta x, \Delta y, \theta)$.
2. Sample transformation space to discrete points.
3. Use 3-D Hilbert scan to convert the 3-D parameter space into 1-D space.
4. Find the optimal parameters in the search space using HSD as the similarity measure.

We use Hilbert scan to convert the 3-D space into 1-D sequence as $H = \{H_1, H_2, \ldots, H_L\}$ where $L$ is the total number of the points in the 3-D parameter space. Each point in the 1-D sequence corresponds to a transformation and has an HSD value. We should find a point whose distance is the smallest. Of course, when the number of the points becomes large, it takes much time to do the full search. Here we will show how to search more efficiently.

Some notions used in the search algorithm are as follows:

- $C_i$: the $i$th candidate point in current candidate set;
- $S = \{C_i|i = 1, 2, 3, \ldots, N\}$: current candidate set including $N$ candidates;
- $M_i$: the middle point of the left or right adjacent intervals of $C_i$;
- $C_{min}$: the candidate point with the smallest HSD value in current candidate set;
- $G_{min}$: the candidate point with the smallest HSD value in whole search procedure;
$HSD(x)$: the directed HSD value of a candidate point $x$;

$F(x)$: a selection function for keeping the candidate $x$ to the next search stage.

The algorithm which is similar to gradient descent is as follows:

**Step 1** Initialize $G_{\min}$ with a large number and equally divide the whole 1-D sequence into $k$ parts using $k + 1$ initial points with interval $\Delta d = \Delta d_0$.

**Step 2** Form $S$ with the $k + 1$ initial points.

**Step 3** For $C_i$, compare its two adjacent points $C_i + \Delta d$ and $C_i - \Delta d$. If $HSD(C_i + \Delta d) \leq HSD(C_i - \Delta d)$, $M_i = C_i + \Delta d/2$; otherwise, $M_i = C_i - \Delta d/2$.

**Step 4** Compare $M_i$ to $C_i$, if $HSD(M_i) \leq HSD(C_i)$, replace $C_i$ with $M_i$.

**Step 5** Repeat Steps 3 and 4 until all candidates are processed.

**Step 6** Find $C_{\min}$ from $S$, if $C_{\min} \leq G_{\min}, G_{\min} = C_{\min}$.

**Step 7** Update $S$ using the selection function $F(x)$.

**Step 8** $\Delta d = \Delta d/2$. If $\Delta d \geq 1$, go to step 3); otherwise, stop and go to next step.

**Step 9** Let point with $G_{\min}$ be the optimal solution.

In Step 7, the selection function is defined as

$$F(x) = \frac{HSD(x) - HSD(G_{\min})}{HSD(x)} - t, 0 \leq t \leq 1$$

(12)

where $t$ is a threshold to control the selection. When $t$ is smaller, fewer candidates will be selected. An extreme case is that when $t$ is 0, only one candidate is selected in each stage. Thus, $t$ gives flexibility between speed and accuracy in the search.

5. Experimental Results and Discussions

In this section, we present the experimental results using our algorithm and analyze the computational complexity. The first experiment is designed for comparing the performance DLSF and WLSF in the first matching stage using both simulated point data and real fingerprint images. The other experiment illustrates the performance of the whole algorithm in real fingerprint matching problem. We also analyze the computational complexity of our algorithm at last.

5.1 Experiment One

We first evaluate the estimated results by using simulated point data. The first column of Fig. 5 shows a point set consisting 250 points generated randomly. We set up two scenarios to generate testing patterns to be matched: a point set by deleting 20% points randomly and a point set by adding 20% points randomly are shown in the second and third columns of Fig. 5, respectively. The fitting results by DLSF and WLSF are painted as the ellipses in Fig. 5. The $r_1$ and $r_2$ are set to 100 and 200, respectively. Then, these ellipses are used to compute the transformation parameters following the proposed method using four axis node points.

![Fig. 5 Fitting results of DLSF and WLSF on simulated data. (a) Results obtained by DLSF. (b) Results obtained by WLSF.](image)

![Table 1 The estimation parameters on simulated data.](image)

Table 1 shows the results of the estimated transformation parameters. We can observe that the estimated parameters by WLSF is much closer to the optimal ones than those by DLSF, and the search space can be reduced to a small space using the estimated parameters.

Fingerprint matching is an important application and most automatic fingerprint identification systems match fingerprint images based on the minutia points [36] as a PPM problem. Figures 6 (a)–(c) show three fingerprint images taken from the same finger at different time with all minutia points extracted. As mentioned previously, we use the upper core point in WLSF as the central point in this experiment. The alignment results by using the estimated transformation parameters from DLSF and WLSF are shown in Fig. 6 (d)–(g). From these figures, it is seen that the results obtained by WLSF are much better than by DLSF, where the core points are much closer or nearly overlapped in alignment results.

5.2 Experiment Two

In this experiment, we use the proposed method for fingerprint matching problem. After minutia points are extracted, we treat the fingerprint matching as PPM problem. Because the fingerprint matching is strongly affected by the minutia extraction and non-rigid deformation (however, we only consider the translation and rotation in this experiment) which often occurs in fingerprint images, the purpose of this experiment is not to show our method is the best method for fingerprint matching, but to show the effectiveness of our method for PPM problem. Suppose we have obtained the best parameters $(\Delta x, \Delta y, \theta)$ using the method we proposed, we can align the two minutia point patterns. Let $P(m_i, m_j)$ be an indicator function that return 1 if the minutiae $m_i$ and $m_j$ are matched by comparing their distance difference $D(m_i, m_j)$ and direction difference $R(m_i, m_j)$:
Fig. 6 Alignment results of fingerprint images (cross means minutiae point and black circle means core point). (a) Fingerprint A. (b) Fingerprint B. (c) Fingerprint C. (d) A to B by DLSF. (e) A to C by DLSF. (f) A to B by WLSF. (g) A to C by WLSF.

\[
P(m_i, m_j) = \begin{cases} 1 & D(m_i, m_j) \leq d_0 \& R(m_i, m_j) \leq \theta_0 \\ 0 & \text{otherwise} \end{cases}
\]

When the function \( P \) gets value 1, we define these two minutia points as matched pair. We also define the matching score \( Q \) to quantize the effect of the matching procedure:

\[
Q = \frac{\#\text{matched pair}}{\max(M, N)}
\]

where \( \#\text{matched pair} \) is the number of the matched pairs, \( M \) and \( N \) are the numbers of the minutia points in template and input images, respectively. So, the fingerprint matching problem can be simplified as if the \( Q \geq T_0 \), then we define the template and input images as matched.

We take a part of FVC2002 database [37] as the test database in our experiment. We use 800 fingerprint images (100 fingers and 8 impressions for each finger) in FVC2002 DB1.a. Each sample in the subset of a finger is matched against the remaining samples of the same finger to compute the False Rejection Rate (FRR), also referred as False Non Match Rate (FNMR). The first sample of each finger in the subset is matched against the first sample of the remaining fingers to compute the False Acceptance Rate (FAR), also referred as False Match Rate (FMR). For more details about the evaluation, please refer to [37]. The computational time of each matching is about 0.5 s by using c++ on a PC with 2.0 Ghz CPU and 1 G memories. The Equal Error Rate (EER) of the experiment is about 14% as shown in Fig. 7.

Some parameters used in our algorithm are listed as follows:

- The search ranges of \( x, y \) translations and rotation are \([-5, 6], [-5, 6], [-\pi \times 5/48, \pi/8]\) with a pre-alignment of the template and the input images using the estimated transformation parameters.
- All fingerprints are cropped to 128 \times 128 pixels, which is also the size of 2-D Hilbert scan for computing HSD value, centered around the core point (if there is no core point, we define the position of a dummy core point as the mean value of minutia point coordinates).
- The 3-D parameter search space are sampled to 12 \times 12 \times 12 points, which is also the size of 3-D Hilbert scan.
- The threshold value \( \tau \) for computing HSD value is 50.
- The number of initial points in 1-D sequence is 9.
- The threshold value \( d_0 \) and \( \theta_0 \) for matching point pairs are 3 and 0.1, respectively.
- The threshold value \( t \) for selecting candidate is 0.5.
The EER of our proposed method may not be comparable to other competitors in the fingerprint authentication, but considering the purpose of this experiment indicated above and we only use the minutiae point information, it shows the effectiveness of our method for computing the transformation parameters in PPM problem.

5.3 Complexity Analysis

The efficiency of our algorithm includes two aspects: one is from computing the similarity measure HSD and the other is from the search strategy.

Hausdorff distance is a commonly used similarity measure for point pattern matching [2]. If M and N are the numbers of minutiae points in the template and the input images and no efficient data structure such as K-D tree for nearest neighbor searching is used, the number of points involved for computing the Hausdorff distance is MN. However, number of points involved for computing HSD is only 2M because only two left and right neighborhoods in 1-D space are required. It is much smaller than MN since always hundreds or thousands points are extracted in an image.

Then, we can analyze the computational complexity of the proposed search strategy as in [16]. Two extreme cases may occur in the search process: first, all Ci's are replaced by Mi's in each stage and we only need to compute HSD value at Mi/s. If C_ave is the average number of candidates selected for each stage, the number of total search points in this case is

\[ N_1 = k + 1 + C_{ave} \log_2(\Delta d_0); \]  

Second, if no Ci is replaced by Mi in each stage and we have to compute HSD value at Ci + \Delta d and Ci - \Delta d, which results in the number of search points becomes

\[ N_2 = k + 1 + 2C_{ave} \log_2(\Delta d_0). \]  

Here, \( \Delta d_0 = \frac{L}{k} \).

Thus, the average number of \( N_1 \) and \( N_2 \) is treated as the final estimation of computational complexity

\[ N_{ave} = k + 1 + \frac{3}{2} C_{ave} \log_2\left(\frac{L}{k}\right). \]  

6. Conclusions and Future Work

In this study, we propose a two-stage matching algorithm for Point Pattern Matching (PPM) using a new ellipse fitting method named Weighted Least Square Fitting (WLSF) and dual Hilbert scans. Compared to the Direct Least Square Fitting (DLSF) method, WLSF shows its robustness to noise and outlier points. The whole algorithm also gives an accurate and fast solution for PPM problem. Future work will aim at solving the weights in WLSF automatically by analyzing the distribution of the point set and extending the algorithm to more real-world applications.

References


Li Tian is a Ph.D. student of Graduate School of Information, Production and Systems at Waseda University, where he received the M.S. degree in 2006. He received his bachelor’s degree from Xi’an Jiaotong University, China in 2003. His current research interests are in image processing, pattern recognition and computer vision. He is a Research Fellow of the Japan Society for the Promotion of Science, and also a student member of the IEEE.

Sei-ichiro Kamata received the M.S. degree in computer science from Kyushu University, Fukuoka, Japan, in 1985, and the Doctor of Engineering degree from the Department of Computer Science, Kyushu Institute of Technology, Kitakyushu, Japan, in 1995. From 1985 to 1988, he was with NEC Ltd., Kawasaki, Japan. In 1988, he joined the faculty at Kyushu Institute of Technology. From 1996 to 2003, he was an Associate Professor in the Department of Intelligent Systems, Graduate School of Information Science and Electrical Engineering, Kyushu University. Since 2003, he has been a Professor in Graduate School of Information, Production and Systems, Waseda University. In 1990 and 1994, he was a Visiting Researcher at the University of Maine, Orono. His research interests include image processing, pattern recognition, image compression, and space-filling curve applications. Dr. Kamata is a member of the IEEE.