Triangulation Based Skeletonization and Trajectory Recovery for Handwritten Character Patterns

Dung Phan[1], In-Seop Na[2], Soo-Hyung Kim[2*], Guee-Sang Lee[2], and Hyung-Jeong Yang[2]
University of Science– Ho Chi Minh City, Vietnam
Department of Computer Science Chonnam National University Gwangju, Korea
[e-mail: dungptmy@gmail.com, ypencil@hanmail.net, gslee@jnu.ac.kr, hjyang@jnu.ac.kr]
*Corresponding author: Soo-Hyung Kim

Received October 11, 2014; revised November 11, 2014; accepted November 18, 2014; published January 31, 2015

Abstract

In this paper, we propose a novel approach for trajectory recovery. Our system uses a triangulation procedure for skeletonization and graph theory to extract the trajectory. Skeletonization extracts the polyline skeleton according to the polygonal contours of the handwritten characters, and as a result, the junction becomes clear and the characters that are touching each other are separated. The approach for the trajectory recovery is based on graph theory to find the optimal path in the graph that has the best representation of the trajectory. An undirected graph model consisting of one or more strokes is constructed from a polyline skeleton. By using the polyline skeleton, our approach accelerates the process to search for an optimal path. In order to evaluate the performance, we built our own dataset, which includes testing and ground-truth. The dataset consist of thousands of handwritten characters and word images, which are extracted from five handwritten documents. To show the relative advantage of our skeletonization method, we first compare the results against those from Zhang-Suen, a state-of-the-art skeletonization method. For the trajectory recovery, we conduct a comparison using the Root Means Square Error (RMSE) and Dynamic Time Warping (DTW) in order to measure the error between the ground truth and the real output. The comparison reveals that our approach has better performance for both the skeletonization stage and the trajectory recovery stage. Moreover, the processing time comparison proves that our system is faster than the existing systems.

Keywords: Handwritten character, Trajectory Recovery, Skeletonization, Polyline Skeleton, Polygonal Approximation.

A preliminary version of this paper was presented at APIC-IST 2014 and was selected as outstanding paper. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (2014-014400) and This research was supported by the MSIP, Korea, under the ITRC support program (NIPA-2014-H0301-14-1014)

http://dx.doi.org/10.3837/tiis.2015.01.022 **ISSN : 1976-7277**

1. Introduction

The drawing order of static handwritten characters can be recoverd from images, and doing so plays an important role in a wide range of applications [1–5], including offline character recognition, character structure analysis, design of character fonts, etc. To this end, many studies have investigated handwritten trajectory recovery methods over the past decades. Such algorithms are generally classified into three main approaches, including local tracing, global tracing, and hybrid tracing methods.

A heuristic rule method was proposed in Ref. [6] for local tracing, and this method applies some heuristic rules for any feature points where some strokes intersect. However, although a local tracing algorithm usually has a low computation cost, it is very sensitive to noise. Ref. [7–16] developed global tracing methods that improve the performance of the trajectory recovery. The recovery problem is formulated as combinatorial optimization problems by using graph theory. Then, such problems are reduced to the Traveling Salesman problem. However, this approach may lead to a computational explosion if the given handwritten image is complex.

Hybrid tracing algorithms have been studied in order to resolve the issues presented by the above methods [17–19]. This approach uses an algorithm that is based on an approach from graph theory, and the algorithm is executed in two steps. First, the graph structure is analyzed at every crossing points on the given handwritten image, and such information is evaluated against template models in order to judge the types of corresponding edges and vertices. Then, the final drawing order is obtained from the labeling information. However, this method depends on local structures of the given image, and smoothest trajectory possible is not guaranteed.

Most approaches that are used for trajectory recovery are based on the skeleton obtained from the handwritten patterns, and skeletonization is one of the most important steps because it directly affects the results for the trajectory. Skeletonization operates using one of three main approaches that are based on pixel elimination [20–21], modeling [22–23], and contouring [24–25]. Pixel elimination and the model-based approach are robust when used with low-resolution images. However, these two approaches can have many spurious results at the junction area. The contour-based method is often used to handle high-resolution images. The existing contour-based methods depend on the medial axis transform (MAT), and in order to obtain better results for the skeletonization, it requires a large number of sample points. Therefore, the complexity of the algorithm increases.

In the past few decades, many contour-based approaches that use triangulation for the skeletonization have been proposed [26]. The skeletonization from the contour analysis depends on the medial axis transformation, and the skeleton is obtained by searching the local symmetry of the edges. The locus of the points of symmetry is extracted as the medial axis, and these axes are then connected in nodes to form the skeleton. The advantage of this approach is that it can generate an accurate skeleton. However, extensive sample points are required in order to estimate the skeleton, and the complexity of the computation increases.

In 2001, Melhi proposed a novel triangulation procedure for thinning hand-written text [27]. In this method, the boundaries of the words are converted into polygons, the polygons are then converted into sequences of non-overlapping triangles, and finally, each triangle is replaced by its skeleton to form a fully connected word skeleton. If an invariance for the rotation and translation is necessary, it can be achieved by rotating each connected component of a word to produce the principal horizontal direction. Then the chain encoding of the boundary is started from the left intersection with the principal axis. The method offers several advantages relative to alternative techniques. Since the operation is based on word boundaries, the time that it takes for the method to complete does not increase rapidly as the resolution of the images increases, unlike methods that iteratively remove outer layers of the pixels to form the pattern object. However, this method was created specifically for cursive handwriting, and it might be useful in the skeletonization of other line-like patterns, such as those that are found in chromosomes, fingerprints, and line drawings. In 2007, Lam and Yam proposed a skeletonization technique that is based on Delaunay Triangulation and Piecewise Bezier Interpolation [28]. The objective of this study was to develop an efficient skeletonization method that was applicable to high-resolution planar shapes. The algorithm is a medial axis transformation approach that assures the skeletal accuracy. Instead of linear interpolation, a Bezier curve is used to connect the inconsecutive sample points. The Bezier curve can approximate the gradient change of the symmetry point locus. Therefore, the proposed method is applicable to skeleton gradient-dependent applications.

In 2011, Morrison used triangle refinement through a constrained Delaunay triangulation (CDT) to improve the skeletonization method [29], and Baga [30] proposed another improvement. The results show a vast improvement in the refinement algorithm, as compared with the unmodified CDT technique. It took roughly twice as much time to skeletonize the image using the refinement algorithm, than the unmodified CDT technique.
In order to improve the performance and to overcome the current drawbacks of the skeletonization and handwritten trajectory recovery, we propose a novel approach for handwritten skeletonization and trajectory recovery that is based on triangulation and graph theory. Our system is composed of two stages, handwritten skeletonization and trajectory recovery. In our approach, we first propose the skeletonization method based on the polygonal contours of the handwritten character. In order to produce cleaner and smoother contours, the handwritten contours are represented using a polygonal contour approximation with a line-fitting algorithm. Then, Delaunay triangulation is used to generate the triangle mesh of the handwritten characters. From the triangle mesh, the skeleton is estimated based on a moving average of the triangles on the mesh. This method not only makes the junction clear, but it also separates the characters that are simply touching, resulting in polyline skeletonization.

The second stage is used for handwritten trajectory recovery. The approach for this algorithm is based graph theory, and it can be used to find the optimal path in the graph that has the best representation for the trajectory. An undirected graph model is then constructed from the polyline skeleton of the handwritten character that consists of one or more strokes. Then, the graph is decomposed into single strokes by searching for the optimal path. The smoothest path with the lowest cost on the graph represents the final trajectory of the handwritten character, and in advance of using the polyline skeleton of the handwritten character, our approach accelerates the process by searching for the optimal path step. The trajectory of the handwritten character can be more accurately estimated by using this new approach for skeletonization and by combining the geometric characteristics into a graph theory-based approach, which is very useful for the recognition task.

We built our own dataset to evaluate the performance of the proposed method, and the dataset includes testing samples and the ground-truth. The dataset contains thousands of handwritten characters and words images that were extracted from five handwritten documents. For our evaluation, first we compared our skeletonization method against Zhang-Suen. The relative improvement of our method was expressed as in improvement in the match between the restored image and the input image. In order to evaluate the trajectory recovery, we made a comparison using a measurement method that includes the Root Mean Square Error (RMSE) and Dynamic Time Warping (DTW). The comparison indicated that our approach had better performance in the skeletonization stage and in the trajectory recovery stage. Moreover, a comparison of the processing time indicates that our system is faster than the system against which it was compared.

The rest of this thesis is organized into three sections. In Section 2, we talk about the background related to our work. Our proposed system is presented in Section 3, and Section 4 presents the results of the performance evaluation. In the final section, we discuss the conclusions and our future works.

2. Background and Related Works

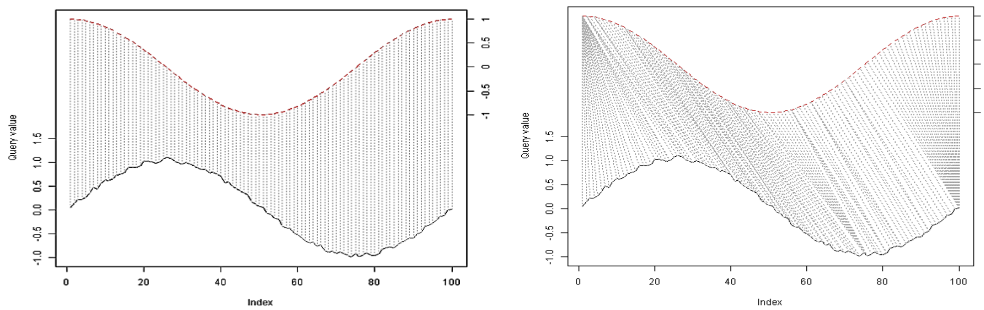
2.1 Delaunay Triangulation

In two dimensions, the triangulation of a set V of vertices is a set T of triangles for which the vertices are collectively V. The vertices in set V have interiors that do not intersect each other, and their union is the convex hull of V if every triangle intersects V only at the triangle's vertices. The Delaunay triangulation D of V, introduced by Delaunay in 1934, is a graph that is defined as follows. Any circle in the plane is said to be empty if it does not enclose a vertex from V. (Vertices are permitted on the circle.) Let u and v be any two vertices of V. A circumcircle (circumscribing circle) of the uv edge is any circle that passes through u and v. Any edge has an infinite number of circumcircles, and the uv edge is in D if and only if there exists an empty circumcircle for uv. The edge that satsfies this property is said to be a Delaunay. Therefore, the Delaunay triangulation of a vertex set is clearly unique because the definition specifies an unambiguous test for the presence or absence of an edge in the triangulation. A Delaunay is every edge that lies on the boundary of the convex hull of a vertex set and has no vertex in its interior. For any convex hull edge e, it is always possible to find an empty circumcircle of e by starting with the smallest circumcircle of e and "growing" it away from the triangulation, and every edge connecting a vertex to its nearest neighbor is a Delaunay. If w is the vertex nearest v, the smallest circle passing through v and w does not enclose any vertices.

It's not at all obvious that the set of Delaunay edges of a vertex set collectively forms a triangulation. For the definition given above, the Delaunay triangulation is guaranteed to be a triangulation only if the vertices of V are in a general position, here meaning that no four vertices of V lie on a common circle. The first step to prove this argument is to describe the notion of a Delaunay triangle. The circumcircle of a triangle is the unique circle that passes through all three of its vertices. A triangle is said to be a Delaunay triangle if and only if its circumcircle is empty.

2.2 Dynamic Time Warping For Time Series Matching (DTW)

A distance measurement between the time series is needed in order to determine the similarity between them and to classify the time series. Euclidean distance is an efficient distance measurement that can be used. The Euclidian distance between the two time series is simply the sum of the squared distances from each nth point in one time series to the nth point in the other. The main disadvantage of using the Euclidean distance for time series data is that its results are very unintuitive. If two time series are identical, but one shifts slightly along the time axis, then the Euclidean distance may reflect them as being very different from each other. Dynamic time warping (DTW) was therefore introduced [34] to overcome this limitation, and it provides intuitive distance measurements between time series by ignoring both the global and local shifts in the time dimension. **Fig. 1** shows an illustration of the matching method of the query and the reference using a normal distance comparison, which provides one-to-one matching, and a DTW-based method, which provides one-to-many matching.



(a) (b)

Fig. 1. Illustration of matching: (a) one to one matching; (b) one to many matching using DTW. DTW looks for the minimum distance mapping between query and reference.

2.2.1 Problem Formulation

The dynamic time warping problem is stated as follows. Given two time series X and Y, in Equation (1), with lengths |X| and |Y|, construct a warp path W defined in equation (2) as

$$X = x_1, x_2, \dots, x_i, \dots, x_{|x|}, Y = y_1, y_2, \dots, y_i, \dots, y_{|y|} \tag{1}$$

$$W = w_1, w_2, \dots, w_i, \dots, w_K, \max(|X|, |Y|) \leq K < |X| + |Y| \tag{2}$$

where K is the length of the warp path and the kth element of the warp path is $w_k = (i, j)$, where i is an index from time series X, and j is an index from time series Y. The warp path must start at the beginning of each time series at $w_1 = (1,1)$ and must finish at the end of both time series at $w_k = (|X|, |Y|)$. This ensures that every index for both time series is used in the warp path. There is also a constraint on the warp path that forces i and j to monotonically increase in the warp path, which is why the lines representing the warp path in **Fig. 1** do not overlap. Every index for each time series must be used, and this can be stated in Eq. (3) as

$$w_k = (i, j), w_{k+1} = (i', j'), i < i' < i + 1, j < j' < j + 1 \tag{3}$$

The optimal warp path is that with the minimum distance, where the distance of a warp path W is defined as in Equation (4) as

$$Dist(W) = \sum_{k=1}^{k=K} Dist(w_{ki}, w_{kj}) \tag{4}$$

$\mathrm{Dist(W)}$ is the distance (typically a Euclidean distance) of warp path W, and $\mathrm{Dist}(\mathrm{w_{ki}}, \mathrm{w_{kj}})$ is the distance between the two data point indices (one from X and one from Y) in the kth element of the warp path.

2.2.2 DTW Algorithm

A dynamic programming approach is used to find this minimum-distance warp path. Instead of attempting to solve the entire problem at once, solutions to sub-problems (portions of the time series) are found and are used to repeatedly find the solutions to a slightly larger problem until the solution is found for the entire time series. A two-dimensional |X| by |Y| cost matrix D is constructed where the value at $\mathrm{D(i,j)}$ is the minimum distance warp path that can be constructed from the two time series $\mathrm{X' = x_1, x_2, \dots, x_i}$ and $\mathrm{Y' = y_1, y, \dots, y_i}$. The value at $\mathrm{D(|X|, |Y|)}$ will contain the minimum-distance warp path between the time series X and Y, and both axes of D represent the time. The x-axis is the time of time series X, and the y-axis is the time of time series Y.

To find the minimum-distance warp path, every cell of the cost matrix must be filled. The rationale behind using a dynamic programming approach for this problem is the following. Since the value at $\mathrm{D(i,j)}$ is the minimum warp distance of the two time series with lengths i and j, if the minimum warp distances are already known for all slightly smaller portions of those time series at a single data point away from lengths i and j, then the value at $\mathrm{D(i,j)}$ is the minimum distance of all possible warp paths for the time series that are one data point smaller than i and j, plus the distance between the two points xi and yj. Since the warp path must either increase by one or stay the same along the i and j axes, the distances of the optimal warp paths one data point smaller than lengths i and j are contained in the matrix at $\mathrm{D(i-1,j), D(i,j-1),}$ and $\mathrm{D(i-1,j-1)}$. So the value of a cell in the cost matrix is defined as in Equation (5):

$$D(i,j) = Dist(i,j) + \min\,[D(i-1,j), D(i,j-1), D(i-1,j-1)] \tag{5}$$

After the entire cost matrix is filled, the warp path must be found from D(1,1) to D(|X|,|Y|). The warp path is actually calculated in reverse order starting at D(|X|,|Y|). A greedy search is performed to evaluate cells to the left, down, and diagonally to the bottom-left. Whichever of these three adjacent cells with the smallest value is added to the beginning of the warp path that has been found so far, and the search continues from that cell. The search stops when D(1, 1) is reached.

3. Proposed Method

There are two main stages in our system including handwritten skeletonization and trajectory recovery. In the first stage, we propose the skeletonization method based on the polygonal contours of the handwritten character. In order to make the contour cleaner and smoother, the contours of handwritten character are represented by polygonal contours approximated by using the line-fitting algorithm. Then, Delaunay triangulation is used to generate the triangle mesh for the handwritten character . From the triangle mesh, the skeleton is estimated according to the moving average of the triangles on the mesh, and this method generates the polyline skeletonization. The second stage involves handwritten trajectory recovery. The approach for this algorithm is based on graph theory to find the optimal path in the graph with the best representation for the trajectory. An undirected graph model is then constructed from the polyline skeleton of the handwritten character consisting of one or more strokes. Then, the graph is decomposed into single strokes by searching for the optimal path. The final trajectory of the handwritten character is represented by the smoothest path on the graph with the minimum cost. The general flowchart of the system is presented in **Fig. 2**.

3.1 Triangulation Procedure based Skeletonization

Our proposed skeletonization method has four main steps: contour extraction, polygonal contour approximation, Delaunay Triangulation-based Triangular Mesh Generation, and polyline skeleton estimation.

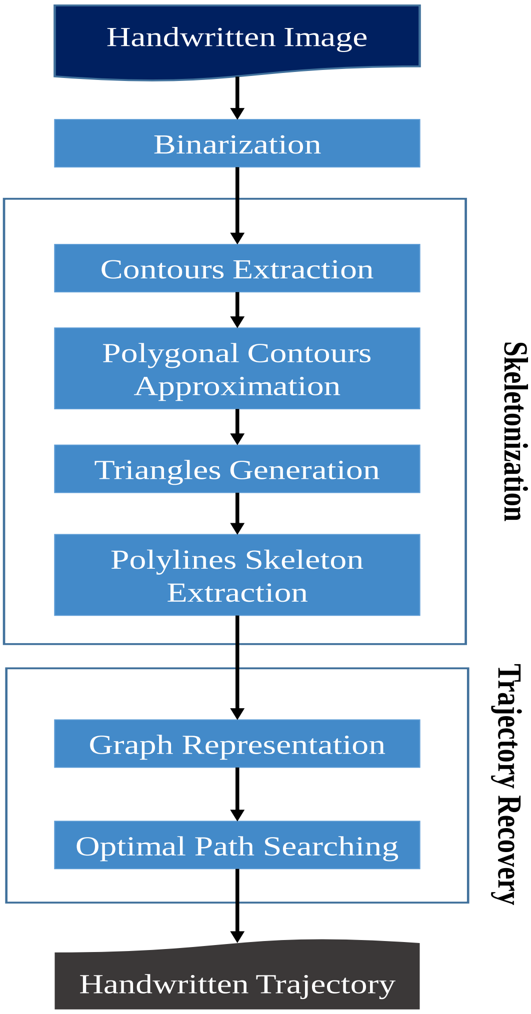


Fig. 2. Flowchart of Proposed System

3.1.1 Contours Estimation and Polygonal Contours Approximation

This step references Dilip K. Prasad's work [31, 32]. He proposed an analytical formula that can be used to choose the optimal threshold value for the line-fitting algorithm, such that the user does not need to prescribe a control parameter, and the line fitting method can automatically choose the threshold value depending on the digital curve itself. **Fig. 3** shows pseudo-code of Prasad's method, and as an example, the approximation of the polygonal contours is shown in **Fig. 4**.

Function DP=L-RDP_max ($\{P_1, P_2, \ldots, P_N\}$)
{ DP=NULL; % DP contains the dominant points
%step 1: line and its parameters
Fit a line l using P_1 and P_N.
For the line, find distance $s = |P_1 P_N|$ and slope m.
Compute $\partial\phi_{\max}$ and $d_{tol} = s\partial\phi_{\max}$ using (10), (13) and (14).
% step 2: maximum deviation
Find deviation $\{d_1, \ldots, d_N\}$ of pixels $\{P_1, \ldots, P_N\}$ from the line l.
Find $d_{\max} = \max\{d_1, \ldots, d_N\}$ and point $P_{\max}$ corresponding to $d_{\max}$.
%step 3: termination/recursion condition
If $d_{\max} \leq d_{tol}$
DP= $\{DP, \quad P_1, P_N\}$
Else
{ DP= $\{DP, \quad \text{LRDP\_max}\ (P_1, P_{\max})\}$.
DP= $\{DP, \quad \text{LRDP\_max}\ (P_{\max}, P_N)\}$.
}
End
Remove redundant points in DP.
Return(DP).
}

Fig. 3. Pseudo-code of the Polygonal Contours Approximation Algorithm [31]

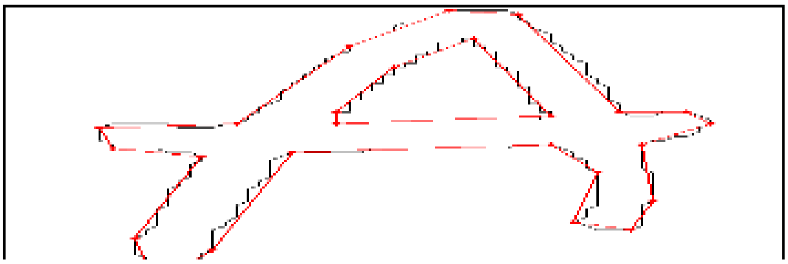


Fig. 4. Example of Polygonal Contours of Handwritten

3.1.2 Triangular Mesh Generation and Constraints for Delaunay Triangulation

The Delaunay Triangulation (DT) is a method that convers a set of Cartesian coordinates into a triangular mesh. It maps circumcircles to vertices, such that each circle intersects the boundary at three vertexes. The vertices mutually connect form a triangle, providing that there is no other vertex inside.

A 2D Delaunay triangulation of a set of points ensures that the circumcircle associated with each triangle contains no other points in its interior. In 2D triangulations, we can impose edge constraints between the pairs of points. In our problem, we define the constraint for the

DT based on the polygonal boundary. In order to produce the fine triangle mesh, we partition the polygon edges into small edges based on the stroke width value. The stroke width is estimated by using the method introduce in Ref. [35], and an example of this step is shown in **Fig. 5**.

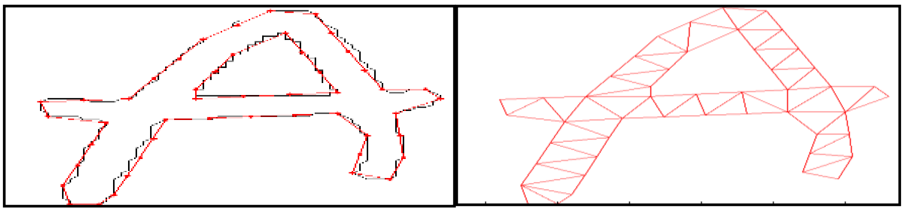
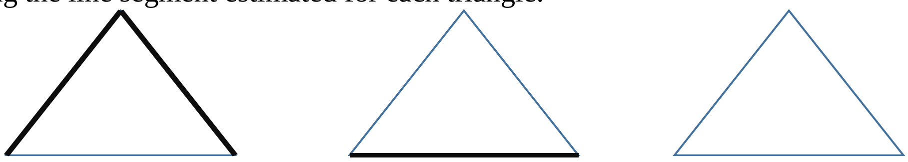


Fig. 5. Example of triangle mesh generation: (a) Red lines defined the constraints for Delaunay triangulation; (b) Triangle Mesh on 'A' character.

3.1.3 Polyline Skeleton Estimation

To estimate the polyline skeleton, we first classify the triangle into 3 classes: terminal, connection, and junction. The terminal triangle contains 2 border edges that are on the polygonal boundary. The Triangle contains 1 border edge that is detected as a connection triangle. The junction triangle is the triangle that does not contain any border edges. See **Fig. 6** for an illustration of the triangle classification. The estimated skeleton is a polyline including the line segment estimated for each triangle.

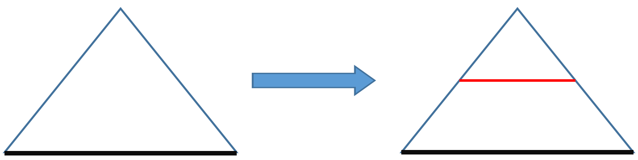


(a) Terminal (b) Connection (b) Junction

Fig. 6. Three types of triangles: Thick lines indicate the edges of the polygon boundary.

(1) Skeleton Estimation for Terminal and Connection Triangles

We consider the terminal triangles as spurious segments, and they are excluded from the skeleton. However, they are used to estimate the line segment s at the junction triangle if they are adjacent. For the connection triangles, the line segments are estimated as the moving average, see **Fig. 7**.

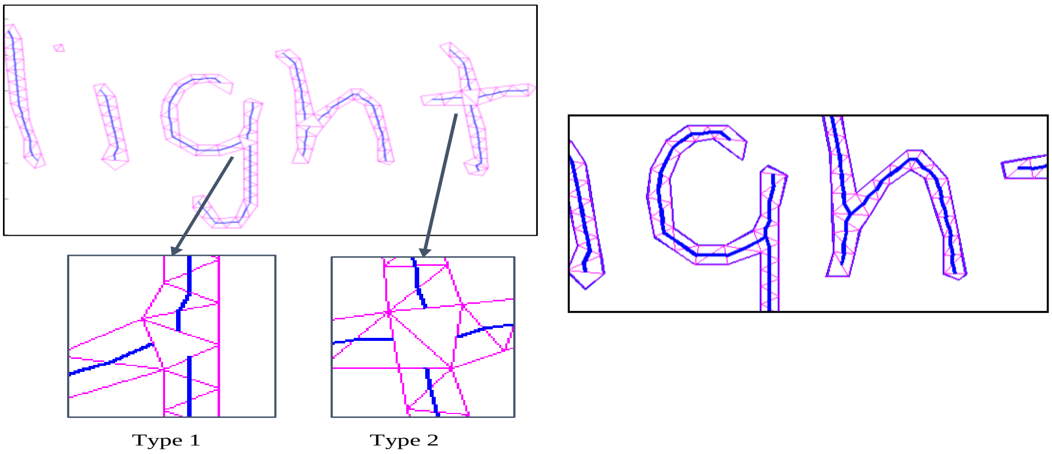


(a) Connection Triangle (a) Estimated Line Segment

Fig. 7. Line segment estimation for the connection triangle: The red line is an estimated line segment.

(2) Skeleton Estimate for Junction Triangle

The most important step in our method is to estimate the line segments for the junction triangle. A handwritten character often has a junction of two lines. However, a junction of three lines, four lines, or more can also appear. First, we solve the problem at a junction of two lines. There are two cases of two-line junctions (**Fig. 8**). For the first type, we search the best couples of the line segment [blue line in **Fig. 8(a)**] which has the smallest direction change. The connection of these two line segments is the estimated line segment. The remaining line is extended forward to the estimated line segment.



(a) (b)

Fig. 8. (a) Two Types of Junction Triangle; (b) Final Polyline Skeleton.

The second problem is the estimation of the line segment for a junction of more than 2 lines. In this case, many junction triangles are adjacent, and therefore we find the convex polygon that contains these adjacent junction triangles. Then, the circumcenter of the convex polygon is extracted, and the estimated line segments are the connections between the middle of each edge of the convex polygon to the circumcenter. **Fig. 8(b)** shows an example of the final skeleton.

3.2 Trajectory Recovery for Handwritten

Our approach for this step is based on the graph theory approach that can be used to find the optimal path in the graph with the best representation of the trajectory. An undirected graph model is constructed from the polyline skeleton of the handwritten character consisting of one or more strokes. Then, the graph is decomposed into single strokes by searching for the optimal path. The final trajectory of the handwritten character is represented by the smoothest path on the graph with the minimum cost. Our approach accelerates the process by searching for the optimal path before using the polyline skeleton of the handwritten character.

3.2.1 Assumption for Trajectory Recovery

In our trajectory recovery approach, we assume that there is no double trace in the handwritten trajectory. Therefore, multiple strokes can be decomposed through an optimal path search. Another assumption for detecting the start point and the end point is that the handwritten trajectory should start from left to right without any redundant trace. For the junction process, our proposed method can handle well with the junction with a limited number of strokes that is intersected at a junction.

3.2.2 Graph Representation of the handwritten skeleton

We define an undirected graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ where $\mathrm{V} = \{\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_\mathrm{n}\}$ indicates a set of vertices, and $\mathrm{E} = \{\mathrm{e}_1, \mathrm{e}_2, \ldots, \mathrm{e}_\mathrm{m}\}$ indicates a set of edges. Each vertex $\mathrm{v}_\mathrm{i} \in \mathrm{V}, \mathrm{i} = 1,2, \ldots, \mathrm{n}$ is a geometrica feature point that corresponds to some terminal point or some intersection of some stroke segment. Each edge $\mathrm{e}_\mathrm{i} \in \mathrm{E}, \mathrm{i} = 1,2, \ldots, \mathrm{m}$ corresponds to a segment of a stroke in a handwritten skeleton.

From the polyline skeleton, we can easily construct a graph representation. We can consider polyline as a graph in which set of edges are set of polyline and set of vertices are a set of points from the polyline. The set of vertices and edges can be normalized by removing unnecessary vertices with a degree of two.

3.2.3 Optimal Path Search For Trajectory Recovery

Whenever we search for an optimal path in the graph, we need to identify the vertex from which we start the trace. It is important to first find the optimal path. Frequently, human writing starts up at a new point, which means that the start point should be a vertex with degree of one. However, a vertex of degree one may be a start/end point, or the start point can touch other strokes.

To select the start point, we look for a candidate from all vertices with an odd degree in the graph. The start point is the most left vertex in a list of odd vertices. In the case where there are near odd vertices around the candidate of the start point, we consider the position of these two points in order to select the best candidate. After the start point is detected, the process for searching the optimal path is performed in order to obtain the final trajectory.

According to the start point, we will begin to search for the optimal path in the graph. The seach algorithm is known as the Basic Trace Algorithm (BTA). Given that a start vertex and one of edges are connected to the vertex, we search for the candidate adjacent edge that has the minimum cost value representing the direction difference between the current edge and the adjacent edge. However, the stroke segments in the handwritten character are always likely to be curve, which may cause a mismatch between the cost value and the smoothness at the connection of the two edges if we use the entire edge for the cost calculation. Therefore, we first estimate a unit vector, which indicates the local direction of an edge. The cost value between e_1, and e_2 is defined in Equation (6) as

$$costValue(e_1, e_2) = cosine(v_1, v_2) = \frac{(x_1 * x_2 + y_1 * y_2)}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2)}} \tag{6}$$

where e_1, e_2 indicates an edge in the graph, and v_1, v_2 indicate unit direction vectors corresponding to e_1, e_2. $v_1 = (x_1, x_2), v_2 = (y_1, y_2)$.

3.2.4 Multi-stroke decomposition through optimal path searching

Based on the optimal path search, a trace (single stroke) is assessed in the graph until an odd vertex is reached. Then, the start point is updated from the list of remaining vertices in the graph. The process used to find an optimal path is looped until all of the edges in the graph are passed. **Fig. 9** shows a result of the trajectory recovery with multi-stroke decomposition.

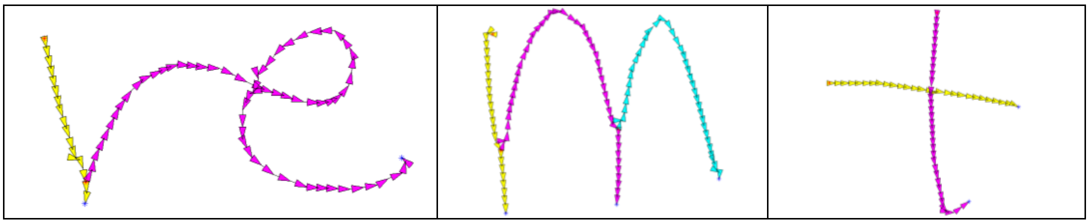


Fig. 9. Example of the decomposition of multiple strokes: each color marks for each stroke.

```
Algorithm 1: Multi-Strokes Decomposition through Optimal Path Searching
Input: Graph (adjacent matrix of graph, list_vertices, list_edges)
Output: list_strokes = List of Optimal Paths
--------------------------------------------------------------------------------
start_point = find start point from list_vertices
current_v = start_point;

while len(list_edges)>0 do
        if mod(degree(current_v),2) && current_v != start_point then
                current_v = find new start point from remain vertices
                list_strokes = add stroke into list of strokes
                stroke = []; % reset stroke
        else
                stroke = add current_v to stroke
                next_v = find next candidate vertex (Algorithm 2)
                Update adjacent matrix
                Remove edge of (current_v,next_v) out off list_edges

                if degree(current_v) == 0 then
                        Remove vertex current_v out off list_vertices
                end if
                prev_v = current_v
                current_v = next_v
        end if
end while
```

```
Algorithm 2: Find next candidate vertex in Graph
Input: Graph, previous and current vertex (prev_v, current_v)
Output: next candidate vertex (next_v)
--------------------------------------------------------------------------------
vec1 = current_v-prev_v
list_adj_v = list of adjacent vertices
max_w = MIN_VALUE
for i=1:len(list_adj_v) do
        v_adj = list_adj_v(i)
        vec2 = v_adj-current_v
        weight = getCost(vec1,vec2) % get cost of 2 vector as equation (2)
        if weight > max_w then
                max_w = weight
                next_v = v_adj
        end if
end for
```

4. PERFORMANCE EVALUATION

4.1. System Environment and Database

Our system was implemented in MATLAB R2012b on our workspace with a system configured with an Intel® Core™ i3-3220 CPU @ 3.30 GHz, 8G RAM, Windows 8.1 64b. Our database was generated from five online documents in the XML format handwritten by five people. The online handwritten documents contain a list of point sequences for each stroke of the character/words in the whole document. In order to evaluate our system, we generate the test and the training data two types of images including single-stroke images and multi-stroke (character/word) images from these documents. The test images are restored from point sequences from the online dataset, and the first dataset includes 2180 images with a single stroke. The second dataset includes 1640 images of multiple strokes, which can be either handwritten characters or words. A ground truth dataset is extracted corresponding to the test image.

4.2 Method for Performance Evaluation

(1) Precision, Recall, and Accuracy for Skeleton Evaluation

In order to measure the accuracy and the error of the output skeleton, we propose a procedure that consists of the following three steps: stroke width estimation [35], image reconstruction from the polyline skeleton with an estimated stroke width, and precision, recall and accuracy calculation with the input image and reconstructed image. The following formulas define the precision, recall, and accuracy computation [see equations (7, 8, and 9].

$$Precision(I,J) = \frac{tp}{tp+fp} \tag{7}$$

$$Recall(I,J) = \frac{tp}{tp+fn} \tag{8}$$

$$Accuray(I,J) = \frac{tp+tn}{tp+tn+fp+fn} \tag{9}$$

where tp,tn,fp, and fn correspond to the true positive, true negative, false positive, false negative, and they are compared to the result of the restored image I (test) and a given input image J (ground truth).

(2) Root Mean Square Error (RMSE) for Handwritten Trajectory Matching

The RMSE is a measure of precision, and it is used to determine the accuracy of the transformation from one coordinate system to another. It is the difference between the desired output coordinate for the ground truth of the trajectory and the actual one. The formula for the RMSE of the point sequences is defined following as Equation (10):

$$RMSE = \sqrt{\sum_{i=1}^{n}\sum_{t=1}^{l_i}(x_t - \hat{x}_t)^2 + \sum_{i=1}^{n}\sum_{t=1}^{l_i}(y_t - \hat{y}_t)^2} \tag{10}$$

where x_t and y_t are the actual coordinates of the signature i at time t; $\hat{x}_t$ and $\hat{y}_t$ are the predicted coordinates of handwritten i at time t; and l_i is the length of handwritten i.

(3) Dynamic Time Warping (DTW) for Handwritten Trajectory Matching

In general, DTW is a method that is used to calculate the optimal match between the two given sequences (e.g., time series) with certain restrictions. The sequences are "warped" non-linearly in the time dimension in order to determine the measure of their similarity in a manner that is independent of certain non-linear variations in the time dimension. The detail for this method was presented in Section 2.3.

In our system evaluation, we apply the DTW method to the trajectory evaluation in order to find the optimal matching between the ground truth point sequences and the real-output stroke point sequences of our system. The DTW measurement provides the minimum error cost possible for the two point sequences. We apply the DTW on both the x-coordinate and y-coordinate to obtain the final error cost between the two point sequences.

4.3 Evaluation of the Skeletonization Method

In order to evaluate our skeletonization method, the datasets were also examined with Zhang-Suen, a state-of-the-art skeletonization method. We estimate the value of the duplicated rate, precision, and recall, as presented in Section 4.2(1). Besides, we also evaluate the processing time in order to show our fast algorithm, even though our system spends time for polygonal contour estimation. According to the report of the evaluation and the comparison in **Table 1**, our proposed method shows a competitive result compared against the Zhang-Suen method. Moreover, in **Fig. 10**, the result of the comparison shows that our proposed method is more robust in the case where there is junction distortion. In the Zhang-Suen result, the skeleton of the characters around the junction are distorted in the 'A, 't', and 'e' characters. These kinds of distorted skeletons may mislead the trajectory recovery process. However, our result provides a good skeleton, which is extremely useful for the next trajectory recovery step.

4.4 Evaluation of Trajectory Recovery

In order to evaluate the trajectory recovery, we developed two systems in order to make a comparison. The first system uses our proposed method, which includes the proposed polyline skeleton estimation and the proposed handwritten trajectory recovery methods. In the second system, we re-implement the graph-based trajectory recovery method that uses the output of the Zhang-Suen skeletization method. We denote it as the 'Zhang + Graph' approach. Two types of errors, explained in Sections 4.3.2 and 4.3.3 are used to evaluate these two systems.

We calculate the average error of these two systems on the two datasets, including the single stroke and multiple strokes. We also measure the processing time for the system in order make a comparison. A report of these two systems is presented in **Tables 2** and **3**. **Table 2** shows the average errors and processing time of the program, and we also calculate the average errors at each pixel to have more detailed information on the errors (**Table 3)**.

Table 1. Comparison of Skeletonization Method

Method	Precision (%)	Recall (%)	Accuracy (%)	Processing Time (ms)
Zhang+Graph	80.43	95.45	94.08	600
Proposed Method	80.89	94.40	94.78	242

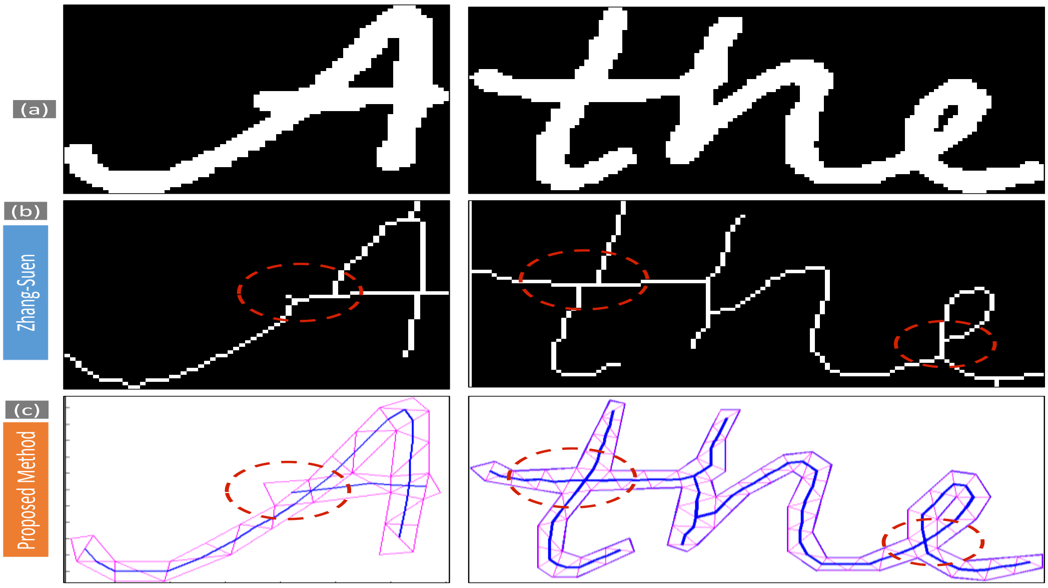


Fig. 10. Experimental Result Comparison: (a) Binary Image; (b) Result of Zhang-Suen Method, around the junction, skeleton of 'A', 't', 'e' characters are distorted.; (c) Result of Our Method, the right skeleton around junction of these character are estimated.

Table 2. Comparison Evaluation: Average Errors and Processing Time per Image.

Data	Method	RMSE	DTW	Processing Time (ms)
Single Stroke	*Zhang+Graph*	688.09	540.94	770
	Proposed Method	559.53	407.8	340
Multi Strokes	*Zhang+Graph*	926.83	714.69	1090
	Proposed Method	787.47	579.25	470

Table 3. Comparison Evaluation: Average Errors per pixel

Data	Method	RMSE	DTW
Single Stroke	*Zhang+Graph*	2.63	2.13
	Proposed Method	1.94	1.5
Multi Strokes	*Zhang+Graph*	2.9	2.35
	Proposed Method	2.18	1.5

The report indicates that our proposed system has better results with a smaller error in the two error measurements, RMSE and DTW. Our system had fewer errors than the 'Zhang + Graph' approach. Moreover, our proposed system is very fast when compared to the other one. The average processing of our system is of 405 ms while the 'Zhang + Graph' takes 930 ms, which is much more than ours. In order to make the observations more clear, we summarize the evaluation in a chart in **Fig.11**. Finally, **Table 4** presents a summary of the evaluation over all the dataset. In the end, the trajectory evaluation has many challenges because human writing is not unique. Therefore, it is not easy to recover the exact handwriting in some cases. **Fig. 12** shows some examples of the handwriting trajectory recovery performed by our proposed system.

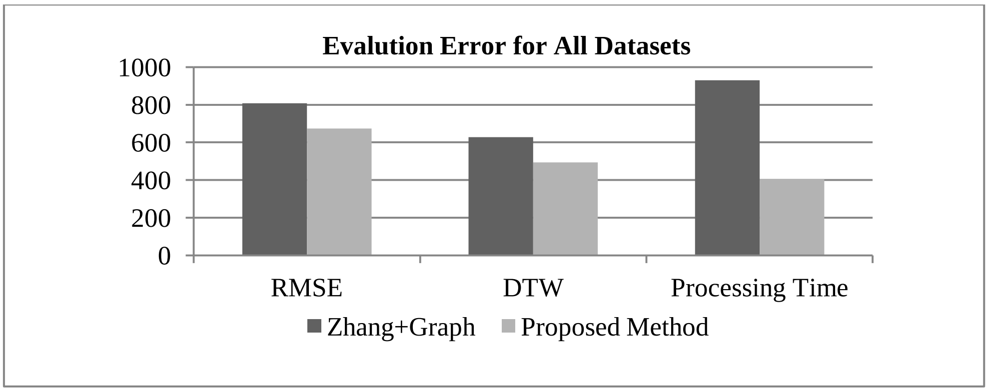


Fig. 11. Performance Evaluation Comparison for All Datasets

Table 4. Performance Evaluation for All Datasets

Method	RMSE	DTW	Processing Time (ms)
Zhang+Graph	807.46	627.815	930
Proposed Method	673.5	493.525	405

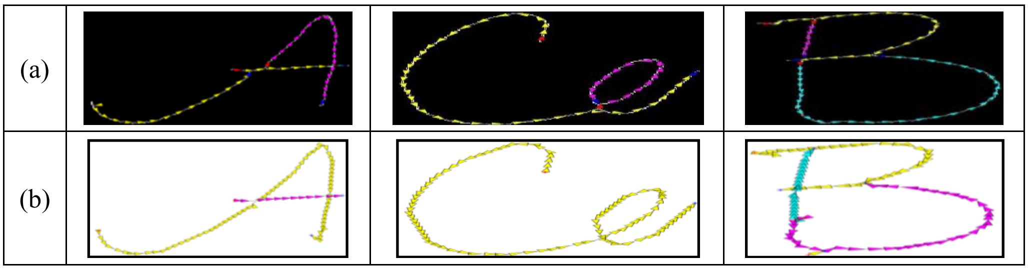


Fig. 12. Experimental Results: (a) Zhang+Graph Based Method and (b) Our Proposed Method

5. Conclusion

In this paper, we propose a novel approach for handwriting skeletonization and trajectory recovery based on triangulation and graph theory. In our approach, we first propose a skeletonization method that is based on the polygonal contours of the handwritten character and a triangulation procedure. This method not only makes the junction clear, but also separates characters that are simply touching. This method generates polyline skeletonization. The second stage recovers the handwritten trajectory. The approach used in this algorithm is based on graph theory and can find the optimal path in the graph with the best representation of the trajectory. The final trajectory of the handwritten character is represented by the smoothest path on the graph with the minimum cost value. In advance of using the polyline skeleton of the handwritten character, our approach accelerates the process by searching for the optimal path. By using this new approach for skeletonization and combining the geometric characteristics into the graph theory approach, the trajectory of the handwritten character can be more accurately estimated, which is very useful for the recognition task. Moreover, the polyline skeleton used for the trajectory and lead trajectory recovery provides a trajectory close to that of human writing. The performance evaluation of the skeletonization and trajectory recovery with the two methods measures the errors of our system and provides a comparison with other methods that show the powerl of our proposed system.

References

[1] Plamondon, R. and S.N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *PAMI, IEEE Trans. on*, 22(1), 63-84, 2000. Article (CrossRef Link).
[2] Srihari, S.N. and E.J. Kuebert, "Integration of hand-written address interpretation technology into the United States Postal Service Remote Computer Reader system," in *Proc. of 4th ICDAR. 1997*, IEEE Press: Ulm Germany. 2, 892-896, 1997. Article (CrossRef Link).
[3] Viard-Gaudin, C., P.-M. Lallican, and S. Knerr, "Recognition directed recovering of temporal information from handwriting images," *Pattern Recognition Letters*, 26, 2537-2548, 2005. Article (CrossRef Link).
[4] Plamondon, R. and C.M. Privitera, "The segmentation of cursive handwriting: an approach based on off-line recovery of the motor-temporal information. Image Processing," *IEEE Transactions on*, 8(1), 80-91, 1999. Article (CrossRef Link).
[5] Babcock, M.K. and J.J. Freyd, "Perception of Dynamic Information in Static Handwritten Forms," *American Journal of Psychology*, 101(1), 111-130, 1988. Article (CrossRef Link).
[6] S. Lee and J. C. Pan, "Offline Tracing and Representation of Signatures," *IEEE Trans. Systems, Man, and Cybernetics*, 22(4), 755–771, 1992. Article (CrossRef Link).
[7] T. Huang and M. Yasuhara, "Recovery of Information on the Drawing Order of Single-Stroke Cursive Handwritten Characters from Their 2D Images," *IPSJ Trans.*, 36(9), 2132–2143, 1995.
[8] S. J¨ ager, "Recovering Writing Traces in Off-Line Handwriting Recognition: Using a Global Optimization Technique," in *Proc. of 13th Int. Conf.on Pattern Recognition*, 931-935, 1996. Article (CrossRef Link).
[9] Kıvári, B., "Time-Efficient Stroke Extraction Method for Handwritten Signatures," in *Proc. of 7th WSEAS International Conference on APPLIED COMPUTER SCIENCE,* Venice, Italy, 2007.
[10] Nel, E.M., J.A. du Preez, and B.M. Herbst, "Estimating the pen trajectories of static signatures using hidden Markov models. PAMI," *IEEE Transactions on*, 27(11), 1733-1746, 2005. Article (CrossRef Link).
[11] Lee, S. and J.C. Pan, "Offline tracing and representation of signatures. Systems," *Man and Cybernetics, IEEE Transactions on*, 22(4), 755-771, 1992. Article (CrossRef Link).
[12] L'Homer, E., "Extraction of strokes in handwritten characters," *Pattern Recognition*, 33(7), 1147-

1160, 2000. Article (CrossRef Link)
[13] Yu Qiao; Nishiara, M.; Makoto Yasuhara, "A Framework Toward Restoration of Writing Order from Single-Stroked Handwriting Image," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11), 1724-1737, 2006. Article (CrossRef Link).
[14] Steinherz, T., et al., "Offline Loop Investigation for Handwriting Analysis," *PAMI, IEEE Transactions on*, 31(2), 193-209, 2009. Article (CrossRef Link).
[15] Niels, R. and L. Vuurpijl, "Automatic Trajectory Extraction and Validation of Scanned Handwritten Characters," in *Proc. of 10th IWFHR. 2006, IRISA/INSA* - Campus Universitaire de Beaulieu: La Baule, France, 2006. Article (CrossRef Link).
[16] Qiao, Y. and M. Yasuhara, "Reccovering Drawing Order From Offline Handwritten Image Using Direction Context and Optimal Euler Path," in *Proc. of 31st ICASSP*. Toulouse, France, 2006. Article (CrossRef Link).
[17] Y. Kato and M. Yasuhara, "Recovery of Drawing Order from Scanned Images of Multi-Stroke Handwriting," in *Proc. of Fifth Int. Conf. on Document Analysis and Recognition*, 261–264, 1999. Article (CrossRef Link).
[18] Y. Kato and M. Yasuhara, "Recovery of drawing order from single-stroke handwriting images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(9), 938–949, 2000. Article (CrossRef Link).
[19] H. Fujioka and T. Nagoya, "A Graph Theoretic Algorithm for Recovering Stroke Order from Multi-Stroke Character Images," in *Proc. of the 2011 2nd Int. Conf. on Innovative Computing and Communication, and 2011 2nd Asia-Pacific Conference on Information Technology and Ocean Engineering*, 34-37, 2014. Article (CrossRef Link).
[20] T.Y.Zhang, C.Y.Suen, "A Fast Parallel Algorithm for Thinning Digital Patterns," *Communication of the ACM*, 27, 236–239, 1984. Article (CrossRef Link).
[21] L. Lam, S. -W. Lee, and C. Y. Suen, "Thinning Methodologies – A Comprehensive Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14, 869-885, 1992. Article (CrossRef Link).
[22] M. Senthilanyaki, S. Veni, K. A. N. Kutty, "Hexagonal Pixel Grid Modeling for Edge Detection and Design of Cellular Architecture for Binary Image Skeletonization," *Annual India Conference (INIDCON)*, 1-6, 2006. Article (CrossRef Link).
[23] R. M. Ralenichka, M. B. Zaremba, "Multi-scale model-based skeletonization of object shapes using self-organizing maps," in *Proc. of 16th Int. Conf. on Pattern Recognition*, 1, 143-146, 2002. Article (CrossRef Link).
[24] H. Rom and G. Medioni., "Hierarchical decomposition and axial shape description," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15, 973-981, 1993. Article (CrossRef Link).
[25] J. J. Zou and H. Yan, "Skeletonization of ribbon-like shapes based on regularity and singularity analyses," *IEEE Trans. on Systems, Man and Cybernetics*, 31, 401-407, 2001. Article (CrossRef Link).
[26] Lakshmi, J.K., Punithavalli, M., "A Survey on Skeletons in Digital Image Processing," in *Proc. of 2009 International Conference on Digital Image Processing,* 260-269, 2009. Article (CrossRef Link).
[27] M Melhi, S.S Ipson, W Booth, "A novel triangulation procedure for thinning hand-written text," *Pattern Recognition Letters*, 22(10), 1059-1071, 2001. Article (CrossRef Link).
[28] Lam, J.H.M.; Yam, Y., "A skeletonization technique based on Delaunay triangulation and piecewise bezier interpolation," in *Proc. of 2007 6th International Conference on Information, Communications & Signal Processing,* 1-5, 2007. Article (CrossRef Link).
[29] Paul Morrison, Ju Jia Zou, "Triangle refinement in a constrained Delaunay triangulation skeleton, Pattern Recognition," 40(10), 2754-2765, 2007. Article (CrossRef Link).
[30] Soumen Bag, Gaurav Harit, "An improved contour-based thinning method for character images," *Pattern Recognition Letters*, 32(14), 1836-1842, 2011. Article (CrossRef Link).
[31] Dilip K. Prasad, Chai Quek, Maylor K.H. Leung, and Siu-Yeung Cho, "Parameter independent line fitting method," in *Proc. of 1st Asian Conference on Pattern Recognition (ACPR 2011)*, 28-30, 2011. Article (CrossRef Link).

[32] Dilip K. Prasad and Maylor K. H. Leung, “Polygonal Representation of Digital Curves,” *Digital Image Processing*, Stefan G. Stanciu (Ed.), InTech, 2012. Article (CrossRef Link).
[33] Paul Morrison, Ju Jia Zou, “Triangle refinement in a constrained Delaunay triangulation skeleton,” *Pattern Recognition*, 40(10), 2754-2765, 2007. Article (CrossRef Link).
[34] Kruskall, J. & M. Liberman, “The Symmetric Time Warping Problem: From Continuous to Discrete,” *In Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison-Wesley Publishing Co., Reading, Massachusetts, 125-161, 1983.
[35] Yoshiyuki Yamashita, Koichi Higuchi, Youichi Yamada, Yunosuke Haga, “Classification of hand-printed Kanji characters by the structured segment matching method,” *In Proceeding of Pattern Recognition Letters 1*, 475-479, 1983. Article (CrossRef Link).
[36] Huy Hoang Nguyen, GueeSang Lee, SooHyung Kim, and HyungJeong Yang, “An Effective Orientation-based Method and Parameter Space Discretization for Defined Object Segmentation,” *KSII Trans. Internet and Information Systems*, 7(12), 180-3199, 2013. Article (CrossRef Link).
[37] Jun Sik Lim, In Seop Na, and Soo Hyung Kim, “Correction of Signboard Distortion by Vertical Stroke Estimation,” *KSII Transactions on Internet and Information Systems*, 7(9), 2312-2325, 2013. Article (CrossRef Link).
[38] Sang Cheol Park, In Seop Na, Tae Ho Han, Soo Hyung Kim, Guee Sang Lee, “Lane detection and tracking in PCR gel electrophoresis images,” *Computers and Electronics in Agriculture*, 83, 85-91, 2012. Article (CrossRef Link).

Dung Phan received her B.S. degree in Computer Science from the University of Science– Ho Chi Minh City, Vietnam in 2009. She recieved the M.S degree from the Department of Computer Science, Chonnam National University, Korea. Her research interests are image processing, pattern recognition and object matching.

In Seop Na received his B.S., M.S. and Ph.D. degree in Computer Science from Chonnam National University, Korea in 1997, 1999 and 2008, respectively. Since 2012, he has been a research professor in the Department of Computer Science, Chonnam National University, Korea. His research interests are image processing, pattern recognition, character recognition, and digital library.

Soo Hyung Kim received his B.S. degree in Computer Engineering from Seoul National University in 1986, and his M.S. and Ph.D degrees in Computer Science from Korea Advanced Institute of Science and Technology in 1988 and 1993, respectively. Since 1997, he has been a professor in the Department of Computer Science, Chonnam National University, Korea. His research interests are pattern recognition, document image processing, medical image processing, and ubiquitous computing.

Guee Sang Lee received the B.S. degree in Electrical Engineering and the M.S. degree in Computer Engineering from Seoul National University, Korea in 1980 and 1982, respectively. He received the Ph.D. degree in Computer Science from Pennsylvania State University in 1991. He is currently a professor of the Department of Electronics and Computer Engineering in Chonnam National University, Korea. His research interests are mainly in the field of image processing, computer vision, and video technology.

Hyung Jong Yang received the B.S., M.S., and Ph. D. degrees from Chonbuk National University, Korea. She is currently an associate professor at the Dept. of Electronics and Computer Engineering, Chonnam National University, Gwangju, Korea. Her main research interests include multimedia datamining, pattern recognition, artificial intelligence, e-Learning, and e-Design.