Published in final edited form as:

Comput Aided Des. 2009 December 1; 41(12): 857–864. doi:10.1016/j.cad.2009.05.005.

Automated Freehand Sketch Segmentation Using Radial Basis Functions

Jiantao Pu and David Gur

Imaging Research Center, University of Pittsburgh, 3362 Fifth Avenue, Pittsburgh, PA 15213

Abstract

Freehand sketching is widely regarded as an efficient and natural way for interaction between computers and humans. We present a robust computerized scheme to automatically segment freehand sketches into a series of components with specific geometric meaning regardless of whether these are generated online or offline. This task is a necessary first step toward sketch understanding. By exploiting the interpolation / extrapolation characteristic of Radial Basis Functions (RBFs), a greedy algorithm consisting of forward and backward operations is proposed for finding the minimum set of segmentation points that can be used to reconstruct with high fitting accuracy freehand sketches in the form of implicit functions. To obtain segmentation points, a simple angle based rule is used to remove “bridging” points that provide a smooth transition between consecutive sketch components. Feasibility of the proposed algorithm is demonstrated by a preliminary performance assessment study using ten computer generated drawings. These experiments show that in this dataset sensitivity of the segmentation was higher than 97.5% with a false positive (FP) rate of approximately 25%. The majority of false positive identifications are located on arc regions where a larger number of segmentation points are needed for reconstruction purposes. The primary contribution of this algorithm is that it transforms an ambiguous problem namely, freehand sketch segmentation, into an implicit function fitting operation. Therefore, this proposed approach has several advantages including independence of the actual sketching activity, and the ability for a satisfactory detection of the transition point between a line and an arc or between two arcs.

Keywords

freehand sketch; segmentation; radial basis functions; greedy algorithm

1. Introduction

As an efficient and natural way to communicate ideas, the potential advantages of freehand sketching have been widely recognized and exploited in many fields. The pen is becoming a standard interactive tool associated with and enabled by a large number of computing devices such as tablet PCs, electronic whiteboards, and personal digital assistants (PDAs). Despite substantial progress in this area since its introduction by Sutherland's Sketchpad [1] in 1963, correct interpretation of the implied intent in sketches, namely, sketch understanding, remains a difficult problem. The primary difficulty arises from the fact that freehand sketches are

Corresponding Author: Jiantao Pu, Ph.D., University of Pittsburgh, 3362 Fifth Avenue, Pittsburgh, PA 15213, Tel: (412)-641-2571, Fax: (412)-641-2582, jip13@pitt.edu.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

informal, ambiguous, and implicit as compared with traditional WIMP (Window, Icon, Menu, and Pointer) user interfaces [2]. An important first step in enabling computers to interpret freehand sketches in a consistent and robust manner is the conversion to a formal representation where two basic problems have to be solved, namely sketch segmentation and primitive (or composite) recognition. These two challenges hinder the development of a sketch-based understanding system [3,4]. In this study, we only focus on sketch segmentation while primitive (or composite) recognition, which determines the type of geometric entity a primitive or a composite represents, is beyond the scope of this paper.

Sketch segmentation parses a continuous stream of pen strokes into a series of constituent primitives that have specific semantics and are generally consistent with human visual cognition. A robust and efficient sketch segmentation algorithm is often crucial for the computer to correctly understand the user's intent and use it as the basis of higher-level symbolic and structural analyses. To our knowledge there is no formal definition for segmentation points of freehand sketches, although Saund [5] pointed out this issue many years ago. This makes it difficult and often confusing to determine whether a segmentation result is reasonable or not. In this paper, we propose a new approach to segmentation of freehand sketches using Radial Basis Functions (RBFs) in a greedy manner. The underlying idea arises from the fact that any two-dimensional curved line with arbitrary geometric shapes can be represented with high fitting accuracy by an implicit function using a limited number of constraint points. Given this property, we approximate the freehand sketch by finding the minimum set of control points in a greedy manner and consider this set to represent a superset of the segmentation points specific to the freehand sketch in question. In other words, missing any of these control points would lead to a large fitting error between the reconstructed result and the original sketch. The proposed segmentation procedure does not require explicit knowledge of sketching speed, curvature, or orientation and could be applied to sketches generated either offline or online. Indeed, the mathematical foundation of this algorithm assures robustness and accuracy of the set of control points as segmentation points for the freehand sketch in question.

In Section II, we review prior work related to sketch segmentation. The sketch representation is described in Section III, and the proposed algorithm is introduced in Section IV. To demonstrate the performance of the proposed algorithm, preliminary experiments that were conducted in our laboratory are described in Section V. Finally, a discussion of the characteristics of the proposed algorithm is provided in Section VI.

II. Related Work

A large number of methods have been proposed to segment (or parse) and recognize sketches. Much of this work assumes that each pen stroke represents a single shape, such as a single line or an arc segment, thus requiring the user to explicitly indicate intended segments. To prevent accumulation of recognition errors, Igarashi et al. [6] developed an interactive “beautification” approach for rapid geometric design in which each stroke is beautified separately in a sequential manner. Landay et al. [7] and Lin et al. [8] used a single-stroke gesture recognizer as a primary recognition tool and thus were not concerned with segmentation per se. Forbus et al. [9] largely avoided the segmentation problem by having the user click a button (mark) when he/she began sketching a primitive and click again (mark) when the primitive in question was completed. However, despite the implicit simplicity, these approaches generally resulted in but a limited “natural interaction” because of the imposed constraints on a user.

To automatically parse sketches, several researchers attempted to take advantage of the interactive nature of sketching. Teh et al. [10] described a dominant point detection algorithm based on a concept termed “region of support”. Given a point, its region of support is computed by measuring its local properties (e.g., curvature). Then a process of non-maxima suppression

is used to determine the actual dominant points in the sketch. Similarly, Kim et al. [11] used curvature as an important criterion during the segmentation procedure. Their proposed approach was intended primarily for closed curves. Rattarangsi et al. [12] proposed a method that located the “corners” of planar curves in a Gaussian scale space by integrating information obtained at multiple scales. A maximum curvature criterion was used in the parsing scheme to identify these “corner” points. This general detection procedure is time consuming and heavily depends on curvature. Lee et al. [13] presented a similar multiscale “corner” detection method based on Wavelet Transform. Despite the increased efficiency and the relative insensitivity to noise, it has limitations and a fraction of the corner points are frequently missed. Sezgin and Stahovich [14,15] proposed a method in which both stroke direction and stroking speed are considered, and used an underlying assumption that extreme speed combined with high curvature typically corresponded to segmentation points. Similar approaches for parsing sketches were used by other investigators [16-18]. By examining both the shape of the stroke and the motion of the pen tip as the stroke progresses, Calhoun et al. [19] found that it was quite natural to slow the pen when making intentional discontinuities. Thus, different segmentations were identified using the speed profile of the pen stroke. However, these speed- and curvature-dependent approaches have to record the sketching activity in real time and perturbations in users' hand motions can lead to changes in curvature, frequently resulting in incorrect segmentation. Moreover, when sketching is performed at a constant speed, a substantial fraction of segmentation points may be missed.

Unlike methods that consider interactive properties during sketching, Hse et al. [20] proposed the use of two types of templates for segmentation. The first is an ordered sequence, consisting of line segments and elliptical arcs. The second consists only of the number of line segments and the number of elliptical arcs associated with a given sketch. Segment points are defined by the templates hence, this selection process ultimately determines the segmentation performance. Due to the wide variety of shapes that are possible during freehand sketching, one cannot assure that the majority of segments are detected correctly when using a fixed number of templates. Considering the fact that circular arcs are widely used in engineering practices, Saund [5] approached the segmentation problem from the perspective of circular arc approximations, where several criteria are used to determine accuracy including goodness of fit, uniqueness, and end abruptness. Shilman et al. [21] presented a Bayesian, visual language based model for sketch segmentation. This statistical model relates observed pen strokes to probabilities of parsing tree nodes. Each time a symbol is added to the scene, only segmentations that are relevant to the specific new symbol in question are added to the parsing tree. However, when using this approach, visual grammar has to be encoded manually and the parser requires a large number of training examples. Sun et al. [22] implemented a locally optimized algorithm based on an Attraction Force Model, which combines adjacent vertices progressively until the maximum attraction force below a predetermined threshold, is reached. After polygon or curve fitting, the sketches are segmented into primitives (e.g., lines, arcs, and ellipses). This approach may be affected by both sketching perturbations and possible inaccurate convergence of the local optimization scheme. Similarly, Shpitalni et al. [23] proposed a conic section fitting approach. In this approach, the computation depends on the number of sampled points, which in turn could affect the quality of “corner” detection. Notowidigdo et al. [24] used heuristic rules and well established image processing techniques to extract higher-level geometric primitives. However, for general purposes it may be difficult to define a limited set of robust rules. Recently, Gennari et al. [25] developed a sketch-based interface for SPICE electric circuit analysis. A parser was used to segment pen strokes into lines and arcs, followed by geometric and domain-specific knowledge that was used to recognize symbols.

III. Freehand Sketch Representation Using Radial Basis Functions (RBFs)

Freehand sketches may be drawn offline or online. Under the offline scenario, these consist of continuous bitmap-like pixels or points: $\{\mathbf{x}_i = (x_i, y_i)\}_{i=1}^n$, where there is no explicit sequence information contained. However, due to the linear appearance of offline sketches, a specific sequence can be obtained by performing a tracing operation. During online sketching, a stroke S is usually composed of a sequence of small line segments rather than an image bitmaps:

$$S = \{(\mathbf{x}_i, \mathbf{x}_{i+1}, t_i) | 0 \leq i \leq n\} \quad (1)$$

where n is the total number of line segments in a single stroke S , and $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ are the two end points of a short segment drawn at time t_i . Accordingly, a sketching activity A is formed by a sequence of strokes:

$$A = \{S_i | 0 \leq i < m\} \quad (2)$$

where m represents the number of strokes.

As related to visual perception, freehand sketches are actually formed by a series of basic geometric entities such as lines and arcs. When a user conveys his/her ideas using a pen, it is not reasonable to assume that each stroke only represents a single geometric primitive. On the contrary, a stroke often encompasses multiple line segments and arcs, thereby representing informal, implicit, and possibly ambiguous properties. As a result, it is often difficult for computerized systems to achieve high performance in terms of sketch understanding.

3.1 Implicit Function Representation

An alternative approach to representing a freehand sketch is to generate a smooth function that passes through all points forming the sketch. Mathematically, this function can be expressed as

$$F(\mathbf{x}_i) = 0, \quad i = 1, 2, \dots, n \quad (3)$$

where the point set $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ forms the actual sketch. The points can be in any dimensional space. In our study, the dimension is 2. Hence, the sketch segmentation problem is reduced to determining a subset of points from the set $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ in a manner that a sketched segment between any pair of neighboring points has a specific geometric meaning.

3.2 Sketch Fitting Using Radial Basis Functions (RBFs)

Fitting implicit curves and surfaces has been studied extensively in recent years [26-31]. The fitting procedure in our proposed approach is based on the minimization of an energy function. In two-dimensional space, one of the simplest energy measures that can be used for this purpose may be represented as a combination of the square of the second partial derivative along each dimension:

$$\begin{aligned}
 E &= \iint \left[\left(\frac{\partial^2 F}{\partial x^2} \right)^2 + \left(\frac{\partial^2 F}{\partial y^2} \right)^2 + 2 \left(\frac{\partial^2 F}{\partial x \partial y} \right)^2 \right] dx dy \\
 &= \int_{\Omega} (F_{xx}^2 + F_{yy}^2 + 2F_{xy}^2) d\Omega
 \end{aligned} \tag{4}$$

where Ω denotes the curve and E represents a measure of “smoothness” in the energy function. Hence, small perturbations in the curve results in large changes in the computed energy function. In other words, minimization of this energy measure leads to a smooth curve. As pointed out by Duchon [32], this type of energy minimization processes can be solved through representation of a curve using RBFs:

$$F(\mathbf{x}) = \sum_{i=1}^n \lambda_i \Phi(|\mathbf{x} - \mathbf{x}_i|) + P(\mathbf{x}) \tag{5}$$

where Φ is the RBF, λ_i are real number coefficients, P is a low degree polynomial, and $\{\mathbf{x}_i\}$ are the points to be fitted. Given a specific application, the RBFs need to be selected properly for the specific problem in question with the proper space dimensionality [29]. In the case of two-dimensional fitting problems, the thin-plate spline $\Phi(\mathbf{x}) = |\mathbf{x}|^2 \log |\mathbf{x}|$ has been widely used [32,33]. It is used in this study for this purpose as well. Other popular choices for RBF include biharmonic or linear spline $\Phi(\mathbf{x}) = |\mathbf{x}|$ as well as triharmonic spline $\Phi(\mathbf{x}) = |\mathbf{x}|^3$ that are primarily used for three-dimensional fitting problems. Gaussian RBF $\Phi(\mathbf{x}) = e^{-d|\mathbf{x}|^2}$, on the other hand, is particularly suitable for neural networks. In the case of two-dimensional curve fitting problems the polynomial part $P(\mathbf{x})$ is often represented in a linear form:

$$P(\mathbf{x}) = \mathbf{c}\mathbf{x} = c_0 + c_1x + c_2y \tag{6}$$

where $\mathbf{c} = (c_0, c_1, c_2)$.

When F passes through all points $\{\mathbf{x}_i\}$:

$$F(\mathbf{x}_k) = \sum_{i=1}^n \lambda_i \Phi(|\mathbf{x}_k - \mathbf{x}_i|) + P(\mathbf{x}_k) = 0 \tag{7}$$

where λ and $\mathbf{c}$ are the unknowns. To avoid a trivial solution (i.e., $\lambda=0$ and $\mathbf{c}=0$), one can sample additional points to assure that $F(\mathbf{x})$ has non-zero values. One simple way to achieve this goal is to sample points along the normal vectors to each of the points. Given a freehand sketch, the normal vector at a point can be simply estimated in the following manner:

$$\begin{aligned}
 \mathbf{n}_i &= ((\mathbf{x}_{i+1} - \mathbf{x}_i) + (\mathbf{x}_i - \mathbf{x}_{i-1})) \\
 &\quad \times \mathbf{z} = (\mathbf{x}_{i+1} - \mathbf{x}_{i-1}) \times \mathbf{z}
 \end{aligned} \tag{8}$$

where ‘ $\times$ ’ denotes a cross product operation, and $\mathbf{z} = (0,0,1)$, and $2 \leq i \leq n-1$. The normal vector at points $\mathbf{x}_1$ and $\mathbf{x}_n$ can be computationally assigned as the normal vectors of their neighboring points, namely, $\mathbf{x}_2$ and $\mathbf{x}_{n-1}$, respectively. Eq. (8) assures that the sampled points

are always located on either the right or the left side of the sketch in question as shown in Fig. 1

When an off-sketch point is located on either the positive or the negative direction of a normal vector, it indicates the right/left side of an object. Thus, one can assign a positive/negative value to each point that can represent either the off-line distance or it can simply be set as a small real number (e.g. 1). This formulation divides the two-dimensional space into two regions, namely positive and negative, and the curve itself is composed of all points with a zero value. This process is similar to the definition of signed distance field, where a space is divided into “outside/inside” regions, and a dividing surface region that constitutes the boundary of a given shape. Assuming a total of k off-sketch points are sampled, Eq. (7) can now be rewritten in a form of $A B = C$:

$$\begin{bmatrix} \Psi_{11} & \Psi_{12} & \cdots & \Psi_{1m} & 1 & x_1 & y_1 \\ \Psi_{21} & \Psi_{22} & \cdots & \Psi_{2m} & 1 & x_2 & y_2 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots \\ \Psi_{m1} & \Psi_{m2} & \cdots & \Psi_{mm} & 1 & x_m & y_m \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 \\ x_1 & x_2 & \cdots & x_m & 0 & 0 & 0 \\ y_1 & y_2 & \cdots & y_m & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \\ c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (9)$$

where $m = n + k$ represents the total number of sampled points, $v_i = 0$ for all the points located on the sketch itself, $v_i = +1$ or -1 represent the points located on either the negative or the positive directions of the computed normal vectors, and $\Psi_{ij} = \Phi(\mathbf{x}_i - \mathbf{x}_j)$ with $i, j \in [1, m]$. Therefore, as had been shown previously [34], for a symmetric linear system, there is always a unique non-trivial solution to Eq. (9).

IV. A greedy algorithm for segmentation

An implicit fitting procedure using RBFs can represent with high degree of accuracy a freehand sketch since the implicit function passes through all points actually forming the sketch. However, the objective is to find those points that play a key role in this representation and with the support of which the original sketch can be reconstructed precisely. Since the segmentation results of freehand sketches are geometrical primitives and these have specific semantic meanings, each pair of segmentation points can be used to infer the primitive they represent for reconstruction purposes. Using this assumption, one can transform the sketch segmentation problem into finding the minimum set of points that can be used to reconstruct the original sketch. For this purpose, we propose a greedy algorithm that consists of both forward and backward operations. The forward operation consists of the following steps:

1. Select a limited set of points located on the sketch as the initial constraints for an implicit curve fitting. In our study, the starting and ending points of each stroke are selected for this purpose and are always regarded as default segmentation points. A third point is randomly selected and added to the list of constraint points. For simplicity, the mid-point of each stroke is selected. To avoid a trivial solution, off-sketch points along the normal vectors of the selected constraint points are also sampled. In theory, an arbitrary number of points could be selected initially and the selection strategy described here is only presented for efficiency purposes.
2. Calculate the implicit function by solving the linear system in the form of Eq. (9) with the selected set of constraint points.

3. Evaluate the fitting error of the reconstructed implicit function at each point of the sketch and add the point with the largest error to the set of constraint points, then recalculate the implicit function.
4. Repeat Steps (2)-(3) iteratively until the largest computed error is smaller than a predefined value ϵ .

At the end of this forward operation, we actually obtain a list of points that approximate the sketch within an error smaller or equal to ϵ . However, due to the random nature of the initial set of constraint points in Step 1, this process could result in the selection of redundant points. In other words, deletion of some of the points as a result of the forward operation may not have an impact on the approximation error. Therefore, a backward operation is also performed. It includes the following steps:

1. Each of the points identified by the forward operation is removed temporarily from the list followed by reconstruction of the sketch with an implicit function using the remaining points.
2. If the error of the implicit function is larger than ϵ , the removed point is restored. Otherwise, it is deleted from the list.
3. Repeat steps 1 and 2 above until the removal of any additional point on the list leads to a larger error.

Following the backward operation, the remaining constraint points represent the minimum set that would approximate the original sketch during reconstruction with an error of ϵ or less. Fig. 2(a) demonstrates the segmentation results after the forward and the backward operations. It shows that the reconstructed curves approximate the sketch in question quite well, despite the use of but a few segmentation points. The selection of the fitting error ϵ in this greedy algorithm largely depends on the perturbation of a sketch. The larger the perturbation is, the larger the value of ϵ would be, and the smaller the value of ϵ is, the more details are captured during reconstruction, leading to a need for additional constraint points. The selection of an appropriate value has to account for the sketching perturbation as well as the details that are needed to be captured in the specific application of interest. During a limited sensitivity assessment of the sketches tested in this study we found that a good balance can be achieved between desired sensitivity levels and false positive identifications can be achieved when ϵ is empirically set at a value of either 2 or 3 pixels.

Although the set of points selected by the greedy algorithm represents the minimum set of segmentation points for sketch approximation (with an error smaller than ϵ), a small number of points that do not “visually” meet a human expected cognition may remain. This is a result of the RBF implicit functions being smooth at each point. To assure that the implicit function transitions smoothly between two neighboring segments and at the same time satisfies the normal constraints at these points, there is a need for the inclusion of several mid-points that act as “transition bridges”. As Fig. 2(a) shows, redundant points typically occur near true segmentation points in order to enable a smooth transition between the segments. To minimize the number of these seemingly “redundant” points, we apply an angle (directional) based rule to all identified points, namely, points $\mathbf{x}_i$ that satisfy the following criterion are discarded:

$$\arccos\left(\frac{(\mathbf{x}_i - \mathbf{x}_{i-1}) \cdot (\mathbf{x}_{i+1} - \mathbf{x}_i)}{|\mathbf{x}_i - \mathbf{x}_{i-1}| |\mathbf{x}_{i+1} - \mathbf{x}_i|}\right) < \vartheta \quad (1 < i < n) \quad (10)$$

where ϑ is a small angle that depends on the perturbation of a sketching activity. Our limited sensitivity assessment when using the sketches in this experiment suggests that in order

to compensate for possible impact of sketch perturbations, θ can be empirically set to values ranging anywhere from 10 to 30 degrees. In the case of a large perturbation, it is difficult to visually determine whether a corner point caused by the perturbation represents a true segmentation point, or not. We note that the starting and ending points of each stroke are always regarded as segmentation points. This rule is similar to the “curvature” approaches incorporated by others [11-15]. However, in our algorithm, the rule is only applied to the set of points after the execution of the forward and backward operations and is applied solely as an optimization procedure. The result of this step is shown in Fig. 2(b).

V. Experimental results

To qualitatively demonstrate the performance of our proposed algorithm, a preliminary user study was conducted. Ten drawings were generated using commercial software AutoCAD with a large reference set of two-dimensional drawings used in engineering practices, and in which the boundaries between geometric primitives are clearly delineated. The connecting points between the geometric primitives on these drawings could be regarded as the “truth” or the “gold standard” (Fig. 3). There are 122 segmentation points in this set of sketches. In the design of these sketches, two primary principles were followed. First, we assumed that sketching activity consists of strokes that can be regarded as sketching “units”. Hence, these drawings were designed in a manner that in principle each sketch could be completed by a single stroke. Second, the drawings were designed to include as many sketching situations likely to occur in practice as possible. For example, there are various geometric constraints between consecutive primitives such as tangent constraints between a line and an arc in Fig. 3(a) and 3(i) as well as between two adjacent arcs as shown in Fig. 3(b) and 3(d)). Identifying optimal segmentation points is a difficult and frequently an impossible task in these cases when curvature based methods are used, because of the naturally smooth transitions. We believe that these ten “single stroke” drawings are sufficiently complex to cover the majority of possible sketching situations in engineering practices and these sketches include all possible constraints that could challenge the quality of the segmentation.

Our experiment included two types of performance assessments. First, we tested the performance of the computerized segmentation against the drawings themselves. In this evaluation, the drawings were converted into the form of images by taking screenshots that were transformed into sequences of points using an inner border tracing algorithm [35]. In this scenario, time information is completely ignored and the converted representation of the sketch can be regarded as drawings generated offline. The segmentation results are shown in Fig. 4 and presented numerically in Table 1. For visual comparison, we provide the reconstruction results obtained by the implicit function fitting processes of the original drawings when using the actual segmentation points. The reconstruction error (s) was set in all cases to within 1 pixel accounting for the fact that these drawings were in a “beautified” form. Sensitivity in this experiment was 99.2% (121/122), namely only one segmentation point in the example presented in Fig. 3(g) was missed. This point should be located on the arc with a small radius. In addition, 30 different points were identified incorrectly as segmentation points (Table 1). Hence, the false positive (FP) rate in this experiment was 24.5% (30/122). One common characteristic was that all false positive identifications as segmentation points were located on arc regions. Interestingly, despite the symmetric characteristic of some of the sketches in term of shapes, the segmentation results are not always symmetric. This limitation may be partially a result of the image processing operations where small perturbations during the conversion of the sketches to discrete representations could lead to different normal vectors estimation as noted in Eq. (10) even when the points are located on the very same arc. In contrast, in cases where there are no arcs (e.g., Fig. 3(f)), there were no false positive identifications.

The second performance assessment experiment of the proposed scheme included true freehand sketches by three users who were asked to draw the same two-dimensional sketches using a computer mouse. Each drawing was drawn using a single stroke. These sketches are expected to approximate the overall “structure” and appearance of the original sketches in terms of shapes but not in sizes or proportions. Thirty drawings were generated for this purpose, namely, three of each sketch. The segmentation and reconstruction performance levels for these freehand drawings are summarized in Table 1 and several examples are shown in Fig. 5 to enable visual inspection of the fittings. The fitting (or reconstruction) error ϵ was set in this experiment to less than 2 pixels. Both sensitivity levels and FP rates are relatively high. For the three users, sensitivity levels were 97.5% (119/122), 98.4% (120/122), and 97.5% (119/122), respectively, with corresponding false positive rates of 27.0% (33/122), 24.5% (30/122), and 22.1% (27/122). As in the first experiment, all the false positive identifications were located on arc regions and there are no false positive identifications in cases where there were no arcs, as shown in Fig. 3(f) and (j) sketched by users #1 and #3. There was only one FP identified in the sketches generated by user #2.

VI. Discussion

We proposed a new freehand sketch segmentation algorithm using implicit function fitting. A greedy procedure was used to determine the minimum set of segmentation points needed to accurately fit sketches with high degree of accuracy. The greedy fitting procedure used here is based on minimization of an energy function as indicated by Eq. (4). During energy function minimization only a set of “greedily sampled” points and their normal vectors are required. Unlike other approaches [10-15], knowledge of curvature and sketching activity such as speed and orientation are not explicitly used. Unlike a number of previously reported approaches that focus primarily on online sketches [14-19], the proposed reconstruction approach is independent of the point sequence, hence, the method should in principle be suited for both online and offline reconstruction of sketches.

The linear system used to compute the implicit function in (Eq. (9)) is not sparse and its size is proportional to the number of the constraints points. Accordingly, the computational complexity (and cost) depends heavily on the size of this linear system. When there is a large number of constraint points that are considered for implicit function fitting computations (e.g., >1000 points), it is difficult to implement the proposed approach in real time on a modern computer due to the computational load required. Fortunately, the greedy procedure used here offers a reasonably fast computation time despite the use of an error based iterative procedure because the number of constraint points in this procedure increases progressively and the final number of constraints is typically reasonably similar to the actual number of segmentation points. In our experiments, the computational time required ranged from 0.3 to approximately 3 seconds given a typical personal computer (AMD Athlon™ 64×2 Dual 2.11 GHz with 2GB of RAM).

The proposed algorithm resulted in high sensitivity. Specifically, the segmentation points at the transition between an arc and a line or between two arcs were detected correctly in the vast majority of instances. However, at the current stage we observed a relatively high false positive rate. As previously pointed, almost all the false positive identifications were located on arc regions. The primary reason is that at minimum of three points may be required for determining an arc correctly while only two points were defined as segmentation points for each arc in our preliminary experiments. We also note that the total numbers of false positive identifications in the two experiments was very similar, despite the fact that one was based on the drawings themselves as a gold standard and the other was based on freehand drawings of the same sketches by different users. These results demonstrate the robustness of our proposed approach. To improve performance in terms of reducing the false positive rates one could potentially use

a primitive/composite recognition approach [36-40]. In this case, each set of three neighboring segmentation points can be evaluated independently to determine whether this limited set could form an arc or a predefined curve. If indeed it can be done, the mid-point can be ignored (discarded) as a segmentation point. However the use of this or other approaches for this purpose is beyond the scope of this paper.

The assessments performed in this study used only 10 reference drawings and 30 freehand sketches. This sample size may not be sufficient for a comprehensive evaluation of all possible situations, albeit we did attempt to cover as many sketching forms that may occur during a single stroke, as possible. The lack of a large, publicly available, reference database of sketches “known exactly” that could serve as a basis for intra and inter scheme comparisons of performance levels had been pointed out by Chetverikov et al. [42]. We recognize the limitation of this work in that it lacks a direct comparison with other methods. Unfortunately we can only infer here in an indirect way on the performance of our approach as compared with other methods. Gennari et al. [25] asked 10 users to sketch a number of electric circuits. These circuits contained on average 9.25 components. Their reported segmentation accuracy ranged from 74% to 90% with an average reported accuracy of 77% across all users. To detect segmentation points, Sezgin et al. [14] combined two classes of information, namely, stroke direction and stroke speed. By testing against ten figures, they reported an overall accuracy of 96%. In [42], Chetverikov et al. compared their scheme with other four methods using a dataset consisting of 8 sketches, where there was no segmentation point located at smooth transition regions and the false positives occurred on arc regions. In contrast, our own testing dataset may be more or less complex. All we can state is that ours contains a large number of smooth transitions between pairs of arcs or between lines and arcs, which are often difficult to detect in freehand sketches, thereby potentially causing a larger number of false negative identifications. While it is clearly impossible and inappropriate to directly compare our results with results reported by others for the dataset included in our assessments, the performance of the proposed method was quite high given the complexity of the dataset used in this study.

Finally, multiple tracing of a stroke for emphasis purpose occurs frequently during sketching [41]. Despite the potential interaction freedom provided by overtracing, this sketch activity will substantially complicate freehand sketch segmentation, and this issue is clearly beyond the scope of the study presented here.

VII. Conclusion

We demonstrated that the problem of freehand sketch segmentation can be approached from the perspective of implicit function fitting and we proposed a greedy algorithm that can find the minimum set of segmentation points needed for accurate fitting of freehand sketches. Our algorithm has several advantages in that it is reasonably robust in detecting smooth transition points and does not rely on explicit computation of the sketching speed or the curvature (orientation). In addition it is independent of orientation and relatively easy to implement. Our preliminary experimental results suggest that the performance of this algorithm is quite promising and it may be further improved by reducing the false positive rate using other approaches.

Acknowledgments

This work is supported in part by grants P50 HL084948 and R01 HL085096 from National Heart, Lung, and Blood Institute, National Institutes of Health, to the University of Pittsburgh. We would like to thank the reviewers and Editor for their comments and constructive recommendations.

References

1. Sutherland, IE. PhD thesis. MIT; 1963. Sketchpad—A Man-machine Graphical Communication System.
2. Li, Y. CS Technical Report. University of California; Berkeley: Oct 22. 2003 Incremental Sketch Understanding for Intention Extraction in Sketch-based User Interfaces. UCB//CSD-03-1284
3. Kara, LB.; Stahovich, TF. Hierarchical Parsing and Recognition of Hand-sketched Diagrams. Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology; Santa Fe (USA). 2004. p. 13-22.
4. Costagliola, G.; Deufemia, V.; Polese, G.; Risi, M. A Parsing Technique for Sketch Recognition Systems. Proceedings of 2004 IEEE Symposium on Visual Languages-Human Centric Computing (VLHCC'04); Rome (Italy). 2004. p. 19-26.
5. Saund E. Identifying Salient Circular Arcs on Curves. CVGIP: Image Understanding 1993;58(3):327–337.
6. Igarashi, A.; Matsuoka, S.; Kawachiya, S.; Tanaka, H. Interactive Beautification: a Technique for Rapid Geometric Design. Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology; Banff (Canada). 1997. p. 105-114.
7. Landay JA, Myers BA. Sketching Interfaces: Toward More Human Interface Design. IEEE Computer 2001;34(3):56–64.
8. Lin, J.; Newman, M.; Hong, JI.; Landay, JA. CHI Letters: Human Factors in Computing Systems. ACM Press; 2000. Denim: Finding a Tighter Fit between Tools and Practice for Web Site Design; p. 510-517.
9. Forbus, K.; Lockwood, K.; Klenk, M.; Tomai, E.; Usher, H. Open-domain Sketch Understanding: The nuSketch Approach. Proceedings of the AAAI Fall Symposium on Making Pen-based Interaction Intelligent and Natural; Washington, D.C. (USA). 2004. p. 58-63.
10. Teh CH, Chin RT. On the Detection of Dominant Points on Digital Curves. IEEE Trans Pattern Analysis and Machine Intelligence archive 1989;11(8):859–872.
11. Kim DH, Kim MJ. A Curvature Estimation for Pen Input Segmentation in Sketch Based Modeling. Computer Aided Design 2006;38(3):238–248.
12. Rattarangsi A, Chin RT. Scale-based detection of corners of planar curves. IEEE Trans Pattern Analysis and Machine Intelligence 1992;14(4):430–449.
13. Lee JS, Sun YN, Chen CH. Multiscale Corner Detection by Using the Wavelet Transform. IEEE Trans Image Processing 1995;4(1):100–104.
14. Sezgin, TM.; Stahovich, T.; Davis, R. Sketch Based Interfaces: Early Processing for Sketch Understanding. Proceedings of the 2001 Workshop on Perceptive User Interfaces; Orlando (USA). 2001. p. 1-8.
15. Stahovich, TF. Segmentation of Pen Strokes using Pen Speed. Proceedings of 2004 AAAI Fall Symposium on Making Pen-based Interaction Intelligent and Natural; Washington D.C. (USA). 2004. p. 152-158.
16. Yu, B.; Cai, SJ. A Domain-independent System for Sketch Recognition. Proceedings of the First International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia; Melbourne (Australia). 2003. p. 141-146.
17. Mackenzie, G.; Alechina, N. Classifying Sketches of Animals Using an Agent-Based System. Proceedings of the 10th International Conference on Computer Analysis of Images and Patterns; Groningen (Netherlands). 2003. p. 521-529.
18. Qin SF, Wright DK, Jordanov IN. On-line Segmentation of Freehand Sketches by Knowledge-based nonlinear thresholding operations. Pattern Recognition 2001;34:1885–1893.
19. Calhoun, C.; Stahovich, TF.; Kurtoglu, T.; Kara, LB. Recognizing Multi-Stroke Symbols. Proceedings of the 2002 AAAI Spring Symposium - Sketch Understanding; Palo Alto (USA). 2002. p. 15-23.
20. Hse, H.; Shilman, M.; Newton, AR. Robust Sketched Symbol Fragmentation Using Templates. Proceedings of the 9th International Conference on Intelligent User Interface; Funchal (Portugal). 2004. p. 156-160.

21. Shilman, M.; Pasula, H.; Russell, S.; Newton, R. Statistical Visual Language Models for Ink Parsing. Proceedings of AAAI Spring Symposium on Sketch Understanding; Palo Alto (USA). 2002. p. 126-132.
22. Sun, ZX.; Zhang, B.; Qiu, QH.; Zhang, LS. A Freehand Sketchy Graphic Input System: SketchGIS. Proceedings of the Second International Conference on Machine Learning and Cybernetics; Xi'an (China). 2003. p. 3232-3237.
23. Shpitalni M, Lipson H. Classification of Sketch Strokes and Corner Detection using Conic Sections and Adaptive Clustering. Trans of ASME Journal of Mechanical Design (Technical brief) 1997;119 (1):131-135.
24. Notowidigdo, M.; Miller, RC. Off-line Sketch Interpretation. Proceedings of AAAI Fall Symposium Series 2004: Making Pen-Based Interaction Intelligent and Natural; Arlington (USA). 2004. p. 120-126.
25. Gennari L, Kara LB, Stahovich TF. Combining Geometry and Domain Knowledge to Interpret Hand-Drawn Diagrams. Computers & Graphics 2005;29(4):547-562.
26. Bajaj C, Ihm I, Warren J. Higher-Order Interpolation and Least-Squares Approximation Using Implicit Algebraic Surfaces. ACM Trans Graphics 1993;12(4):3270347.
27. Keren D, Gotsman C. Fitting Curves and Surfaces With Constrained Implicit Polynomials. IEEE Trans Pattern Analysis and Machine Intelligence 1999;21(1):31-41.
28. Fitzgibbon A, Pilu M, Fisher R. Direct least square fitting of ellipses. IEEE Trans Pattern Analysis and Machine Intelligence 1999;21(5):476-480.
29. Li Q, Wills D, Phillips R, Viant WJ, Griffiths JG, Ward J. Implicit Fitting Using Radial Basis Functions with Ellipsoid Constraint. Computer Graphics Forum 2004;23(1):55-69.
30. Carr, JC.; Beatson, RK.; Cherrie, JB.; Mitchell, TJ.; Fright, WR.; McCallum, BC.; Evans, TR. Reconstruction and Representation of 3D Objects with Radial Basis Functions. ACM SIGGRAPH; 2001; Los Angeles, CA. 2001. p. 67-76.
31. Dinh, Huong Quynh; Turk, G.; Slabaugh, G. Reconstructing Surfaces by Volumetric Regularization Using Radial Basis Functions. IEEE Trans Pattern Analysis and Machine Intelligence 2002;24(10): 1358-1371.
32. Dunchon, J. Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces. In: Dolb, A.; Eckmann, B., editors. Constructive Theory of Functions of Several Variable, Lecture Notes in Mathematics. Springer-Verlag; 1977. p. 85-100.
33. Powell, MJD. Tabulation of thin plate splines on a very fine two-dimensional grid. In: Braess, D.; Schumaker, LL., editors. Numerical Methods in Approximation Theory. Birkhäuser-Verlag (Basel); 1992. p. 105-210.
34. Turk, G.; O'Brien, JF. Shape Transformation Using Variational Implicit Functions. ACM SIGGRAPH; 1999; Los Angeles, CA. 1999. p. 335-342.
35. Cormen, TH.; Leiserson, CE.; Rivest, RL.; Stein, C. Introduction to Algorithms. Second. MIT Press and McGraw-Hill; 2001.
36. Sezgin, TM.; Davis, R. HMM-Based Efficient Sketch Recognition. Proceedings of the 10th International Conference on Intelligent User Interfaces; San Diego (USA). 2005. p. 281-283.
37. Hse, H.; Newton, AR. Sketched Symbol Recognition Using Zernike Moments. Proceedings of 17th International Conference on Pattern Recognition (ICPR'04); Cambridge (UK). 2004. p. 367-370.
38. Fonseca, MJ.; Pimentel, J.; Jorge, JA. Cali-an Online Scribble Recognizer for Calligraphic Interfaces. Proceedings of the 2002, AAAI Spring Symposium on Sketch Understanding; Palo Alto (USA). 2002. p. 51-58.
39. Pu J, Ramani K. Implicit Geometric Constraint Detection in Freehand Sketches Using Relative Shape Histogram. EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling. 2007
40. Kara, LB.; Stahovich, TF. An Image-Based Trainable Symbol Recognizer for Sketch-Based Interfaces. Proceedings of AAAI Fall Symposium Series 2004: Making Pen-Based Interaction Intelligent and Natural; Arlington (USA). 2004. p. 99-105.
41. Ku, DC.; Qin, SF.; Wright, DK. Interpretation of Overtracing Freehand Sketching for Geometric Shapes. Proceedings of the 14th International Conference on Computer Graphics, Visualization and Computer Vision; January 2006; UNION Agency - Science Press; 2006. p. 263-270.

42. Chetrikov D, Zsabo S. A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves. Proceedings of the 23rd Workshop of the Australian Pattern Recognition Group 1999:175–184.

Fig. 1.
Two examples of the normal vector estimation at each point on the sketch. Each normal vector is shown as a rod-like short line.

Fig. 2.

Segmentation results: (a) The minimum set of segmentation points after the greedy algorithm with a fitting error (ϵ) of 2 pixels; and (b) the reconstruction result after the angle based rule is applied. The wider curves (black) indicate the original sketches, the narrower curves (green) indicate the reconstructed sketches from the minimum set of segmentation points using the RBFs. The crosses indicate the identified segmentation points, and the arrows point to the locations where actual segmentation points should be located.

Fig. 3.

Ten drawings (sketches) used as a “gold standard” in the performance evaluation study. All ten drawings could be completed by a single stroke. The dots represent the true segmentation points.

Fig. 4. Segmentation and reconstruction results of the ten drawings used as a “gold standard”. The wider curves (black) indicate the fitted drawings, and the narrower curves (green) indicate the reconstructed curves from the segmentation points identified by the greedy algorithm. The crosses indicate the scheme identified segmentation points. Reconstruction errors were less than 1 pixel.

Fig. 5.

Examples of the freehand sketch segmentations. The wider curves (black) were actually sketched by the users, and the narrower curves (green) indicate the reconstructed sketches using the segmentation points identified by the greedy algorithm (marked as crosses). The reconstruction errors were set at less than 2 pixels.

Table 1

Performance assessments of the sketch segmentations of the “gold standard” and the frechand drawings made by the three “users”.

Drawing ID*	Gold Standard		User 1		User 2		User 3	
	TP	FP	12	FP	TP	FP	TP	FP
a (12)	12	5	12	5	12	4	12	5
b (12)	12	2	4	2	12	6	11	1
c (4)	4	2	12	5	4	2	4	4
d (12)	12	4	12	8	12	6	12	6
e (12)	12	4	22	2	12	2	12	2
f (23)	23	0	8	0	23	1	23	0
g (10)	9	0	10	0	8	0	8	0
h (10)	10	3	10	2	10	3	10	4
i (10)	10	10	17	6	10	3	10	3
j (17)	17	0	12	3	17	3	17	2
Sensitivity	99.2% (121/122)		97.5% (119/122)		98.4% (120/122)		97.5% (119/122)	
FP Identification Rate	24.5% (30/122)		27.0% (33/122)		24.5% (30/122)		22.1% (27/122)	

* The number in the first column indicates the actual number of segmentation points in each of the original drawings.