

# A Method to Segment Strokes in Hand-drawn Sketches

Tuanfei Wang<sup>1, a)</sup>, Yanping Hu<sup>1, b)</sup>

<sup>1</sup>Dalian University of Technology, Liaoning 116024, China.

<sup>a)</sup> wtf1437366527@163.com

<sup>b)</sup> hypok@dlut.edu.cn

**Keywords:** Stroke Segment; Natural Interfaces; Tangent Vertices Detection; Curvature

**Abstract.** In this article we present an approach based the type of curve for detecting tangent vertices. First, the geometric features is used to determinate the curve type of the sub-stroke after corners extraction. Second, approximated piecewise parametric curves are obtained and the analytic curvature is token as the extraction principle of tangent vertices between straight line and curves when the vector product and the discrete curvature are applied to find tangent vertices between curves. The expeirmental result shows that our approach achieves a better accuracy in finding tangent vertices.

## 1. INTRODUCTION

The sketch is the first stage of the field of industrial design, especially in the conceptual design, it is also as a direct and natural way of thinking [1]. On one hand, although the traditional pen-paper interaction cannot hinder designers' inspiration and fully show designers' intention, but go against being further modified and save; On the other hand, the current popular CAD system supporting sketch design such as PRO/E, SolidWorks, UG, are using WIMP way and require users to learn in advance, which interrupts the designers' inspiration and is not conducive to creative work.

Sketch recognition is composed of online and offline recognition. The latter mainly extracts the sketch features by using the image processing algorithms, which has been researched maturely for the moment. Online sketch recognition is mainly consist of pretreatment, corner finding, stroke segmentation, primitive recognition and so on. Among this, stroke segmentation is the key step [2, 3], which is also the current research hotspot. Kim and Kim [4] proposed a curvature metric for corner finding. After resample the raw input, curvature calculation is as the direction change at a given point. Wolin et al. [5] presented ShortStraw, introducing the concept of "straw" which indicates the local curvature of each point. All points with a straw value below a given threshold are considered candidate corners. However, this technique is only suitable to polyline corner finding. Xiong and La Viola [6] developed iStraw, which improves the ShortStraw algorithm by including curve detection. IStraw achieves better accuracy than ShortStraw and is able to handle curve and arc segments when maintaining latter computational complexity. Pu and Gur [7] used radial basis functions to segment sketches, which is able to find smooth transition points, but it still remains very high false positive rate. Albert et al. [8] proposed a method called Tangent and Corner Vertices Detection (TCVD), which uses the radius function to find tangent vertices. TCVD is less susceptible to noise than the traditional curvature computation, but it is dependent on the scale which make it be confused between large radius curves and straight lines. Herold and Stahovich [9] developed ClassySeg, which employs machine learning technique to infer the corner points. A statistical classifier is used to identify which candidate segment windows contain true segment points. ClassySeg is general and extensible but need to train a unique for every field.

We present a sketch segmentation method combing geometric features and polynomials fitting. First, the input stroke sequence are preprocessed to reduce the noise. Second, the curvature of any point in the sequence of resampling points is calculated in order to detecting corner vertices. Finally, primitive fitting is used to identify whether the sub-stroke contains tangent vertices and the corresponding algorithm will be applied to deal with. Experiment show our algorithm can effectively segment the hand-drawn sketch.

## 2. HAND-DRAWN SKETCH SEGMENTATION

### 2.1 Pretreatment

We define strokes as the collection of sampling points during the left mouse button press-move-bounce. Size of the collection is different due to the difference of drawers' speed, we resample the original point sequence in equidistance (the distance between resampling points is 8 in this paper) in order to eliminate the effect of speed on the following processes. At the same time, the shake of hand when moving the mouse and sampling device error will introduce noise. In order to de-noise, the weighted mean filtering algorithm has been carried out:

$$p_i = (p_{i-1} + 3 \times p_i + p_{i+1}) / 5 \quad (1)$$

Where  $p_i$  is a point in the collection of resampling points, the size of the sliding window is 5.

### 2.2 Detection of Corner Vertices

Input: the sketched strokes as a collection of resampling points.

Output: a set of corner vertices obtained by approximately curvature estimation for every input point.

Short description: first, the direction change at a point is calculated. After, the discrete curvature at each point is obtained by the rate of the direction change with chord length.

Implementation: the Euclidean distance  $d_i$  of two adjacent point is computed as the expression  $d_i = \|p_i - p_{i-1}\|$ , then the direction change  $\alpha_i$  of resampling point  $p_i$  can be presented in (2). The approximately discrete curvature  $C_i$  at point  $p_i$  can be calculated approximately from the rate of change of  $\alpha_i$  with chord length in (2).

$$\alpha_i = \arcsin \frac{p_i \cdot x - p_{i-1} \cdot x}{d_i} \rightarrow C_i = \frac{|\alpha_i - \alpha_{i+1}|}{d_i + d_{i+1}} \quad (2)$$

As we can see from Fig.1 (b), if the discrete curvature of point  $p_i$  exceeds a heuristic threshold  $C_{min}$  by 1.3 and is the local maximum value of its environment, this point can be added to the candidate corner sets.

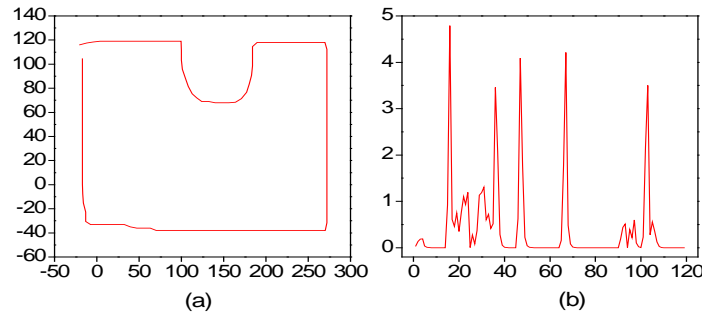


FIGURE 1. (a) Original sketch. (b) Discrete curvature for left sketch

Among the initial corner sets, there may be false corners. For the  $i$ th corner point, if the adjacent corners can be judged collinear and the difference of the slope is not exceed the given threshold, it is considered that the corners are false corners and will be removed from the set of corners. For the false corners in arcs, we draws on the methods proposed by Xiong et al. After the above-mention processing, the fitting of basic primitives are implemented for two adjacent corners when traversing the whole corner sets. If the fitting error is within the given threshold, the sub-stroke is recognition completely, on the contrary, the extraction of the tangent point will be implemented.

### 2.3 Detection of Tangent Vertices

Input: a vector of resampling points between two adjacent corner points where just contain the tangent vertices.

Output: the vector of entities also containing the tangent vertices.

Short description: the resample points between pairs of corners are smoothed by a Gaussian filter to further reduce the noise. Then the type of tangent vertices can be identified by means of

geometric features, if the tangent vertices exist between curves and straight lines, the vector product and the discrete curvature are used to segment strokes. Otherwise, piecewise parametric curves approximate the points between pairs of corners and analytic curvature is regarded as the principle of tangent vertices detecting.

Implement: the resampling points between pairs of corner vertices are smoothed with a Gaussian filter (3), where Gaussian template size  $\sigma$  determines the effect of Gaussian smooth, the corner information will be lost if the size is too large. We take a value of  $\sigma = 8$  by the experimental test:

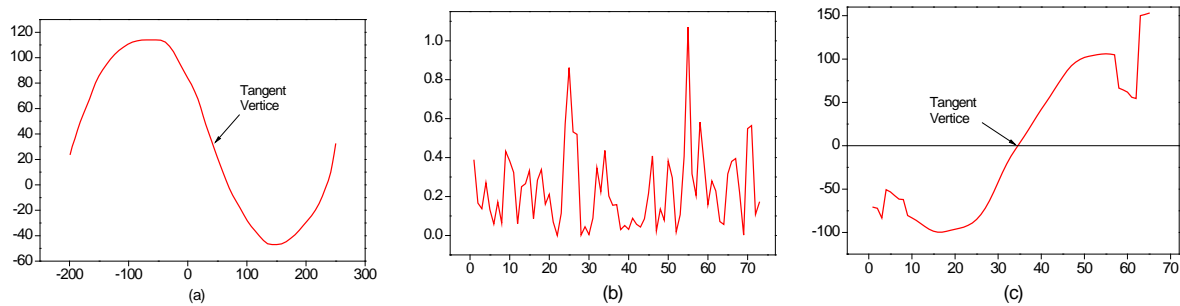
$$g(h) = e^{-\frac{h^2}{2\sigma^2}}, h \in [-\sigma, \sigma] \quad (3)$$

The filtered points  $g(gx_i, gy_i)$  are computed by the convolution of the resampling points with Gaussian filter (4)

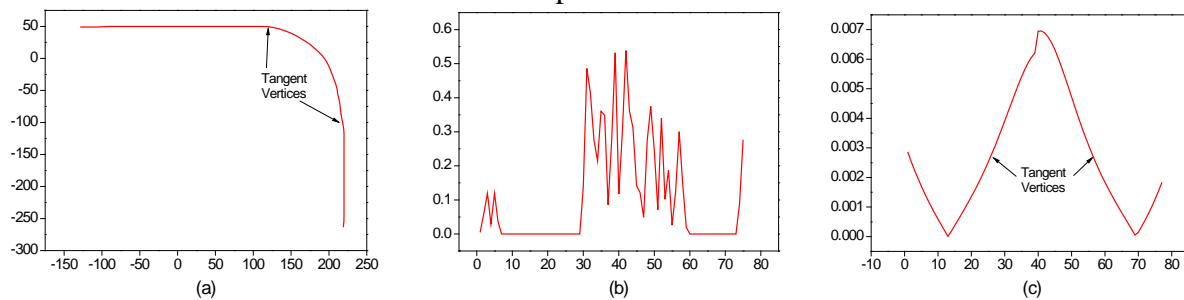
$$gx_i = \sum_{h=-\sigma}^{\sigma} g(h)x_{i+h} / \sum_{h=-\sigma}^{\sigma} g(h), \quad gy_i = \sum_{h=-\sigma}^{\sigma} g(h)y_{i+h} / \sum_{h=-\sigma}^{\sigma} g(h) \quad (4)$$

The tangent points of hand-drawn sketch always exist between arc and straight or between arcs. In order to identify the type of curves, we initialize a threshold  $c_{min}$  by 0.1 of discrete curvature and a ratio threshold  $r$  by 0.4, if the ratio of discrete curvature whose value are larger than  $c_{min}$  is below  $r$ , then the stroke can be identified to be a curve of arc and straight line, contrarily a curve of arc and arc.

For internally-tangent arc, the curvature value at the tangent point changes dramatically. Therefore, the tangent vertices exist where the difference between the discrete curvature values of adjacent points is maximum. For externally-tangent, as Fig 2 shows, the outer product (see Fig.2 c) have better stability than the discrete curvature (see Fig.2 b), and the sign of the vector product will change at the tangent vertices. First, we set the window size  $K$  by 5 for the outer product operation. After, the vector product  $w_i$  for three adjacent points  $p_{i-k}, p_i, p_{i+k} (k < i < n - k)$  is calculated as:  $w_i = ((X_i - X_{i-k})(X_{i+k} - X_i) + (Y_i - Y_{i-k})(Y_{i+k} - Y_i))/k$ . Then if  $w_i > 0, w_{i+1} < 0$  or  $w_i < 0, w_{i+1} > 0$ ,  $p_i$  is identified the tangent point.



**FIGURE 2.** (a) Strokes with tangent vertices between arcs. (b) Discrete curvature. (c) Vector product.



**FIGURE 3.** (a) Strokes with tangent vertices between arc and straight line. (b) Discrete curvature. (c) Analytic curvature.

After the strokes are judged to the type of curve between straight and arc, the discrete curvature (as shown in Fig.3 b) is disordered due to the randomness of the users' drawing, on the contrary, the fitted curve can approximately denote function relation among the discrete points set coordinates.

Formulation expression can not only fully reflect the law of data but also facilitate the following calculation of analytic curvature. As shown in Fig.3 c, the analytic curvature is more stable and has certain extent continuity. In this paper, parametric cubic polynomials are used to segment the strokes. The polynomial equation is as the following:

$$\begin{cases} x(t) = a_x + b_x t + c_x t^2 + d_x t^3 \\ y(t) = a_y + b_y t + c_y t^2 + d_y t^3 \end{cases}, t \in [0, 1] \quad (5)$$

Where a, b, c, d are the coefficients, t is the parameter corresponding to every resampling point. The method of the parameterization of the accumulated chord length is used to parameterize the data points: let us define  $d_0 = 0$  be the distance of the first data point, the Euclidean distance between the following point  $g_i$  and its previous point  $g_{i-1}$  is  $d_i = \|g_i - g_{i-1}\|$ , the accumulate distances are obtained by the equation  $d_j = d_0 + d_1 + \dots + d_j$ , then the value of parameter t is initialized to  $t_j = d_j/d_{n-1}$ , where n is the total number of data points.

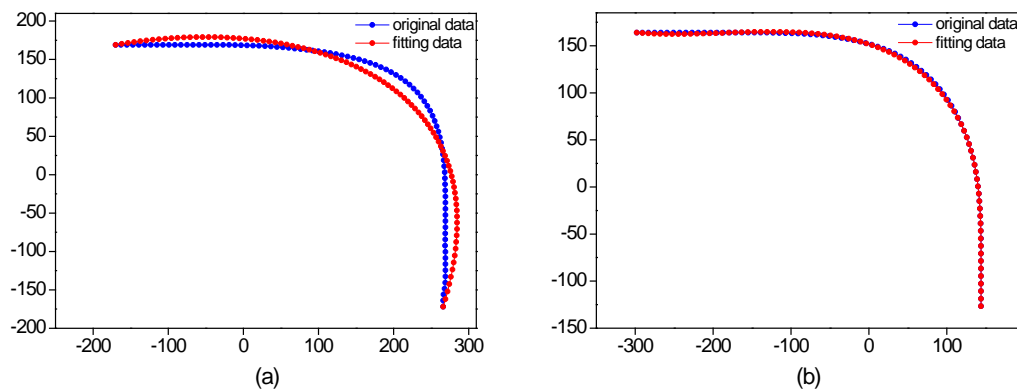
The least square method is used to piecewise parametric cubic polynomials approximation. The polynomials curves are constrained to pass the initial point and the final point:

$$\begin{aligned} t = t_0 &\rightarrow \begin{cases} a_x + b_x 0 + c_x 0^2 + d_x 0^3 = xp_0 \rightarrow a_x = xg_0 \\ a_y + b_y 0 + c_y 0^2 + d_y 0^3 = yp_0 \rightarrow a_y = yg_0 \end{cases} \\ t = t_{n-1} &\rightarrow \begin{cases} a_x + b_x 1 + c_x 1^2 + d_x 1^3 = xp_{n-1} \rightarrow b_x = xg_{n-1} - a_x - c_x - d_x \\ a_y + b_y 1 + c_y 1^2 + d_y 1^3 = yp_{n-1} \rightarrow b_y = yg_{n-1} - a_y - c_y - d_y \end{cases} \end{aligned} \quad (6)$$

The left data points and the result from Eq. (6) are substituted in Eq. (5):

$$\begin{bmatrix} (t_1^2 - t_1) & (t_1^3 - t_1) \\ (t_2^2 - t_2) & (t_2^3 - t_2) \\ \dots & \dots \\ (t_{n-2}^2 - t_{n-2}) & (t_{n-2}^3 - t_{n-2}) \end{bmatrix} \begin{bmatrix} c_x \\ d_x \end{bmatrix} = \begin{bmatrix} xg_1 - xg_0 + xg_0 \cdot t_1 - xg_{n-1} \cdot t_1 \\ xg_2 - xg_0 + xg_0 \cdot t_2 - xg_{n-1} \cdot t_2 \\ \dots \\ xg_{n-2} - xg_0 + xg_0 \cdot t_{n-2} - xg_{n-1} \cdot t_{n-2} \end{bmatrix} \quad (7)$$

After obtaining the result of Eq. (7) by means of the least square method, then all the coefficients are substituted to Eq. (5) in order to calculate the coordinates of fitting data points (see Fig.4).



**FIGURE 4.** (a) 1 parametric cubic curves approximation. (b) Piecewise parametric cubic curves approximation.

The above process will not be terminated until the distance from every fitting point to the original point does not exceed a given threshold by 5. If the distance is larger, the sequence is halved at the point where the distance is maximum and the process is subsequently applied to the two sub-strokes recursively.

After the cubic curve approximation, the analytic curvature of each point can be denoted by means of the derivation of parametric curve equation

$$c(t) = \frac{|x'(t)y''(t) - x''(t)y'(t)|}{(x'^2 + y'^2)^{3/2}} \quad (8)$$

Where  $x', x''$  are the first and the second derivatives of the abscissa when  $y', y''$  are obtained by the first and second derivatives of the ordinate. If the analytic curvature at that point is below a given threshold by 0.0025, then the point is added the set of tangent vertices. After that, the tangent points merging strategy is applied to remove the false tangent points in order to achieve the segmentation between straight line and arc.

### 3. EXPERIMENTAL WORK

In order to verify the effectiveness of the algorithm, we implement several experiments using the mouse as a input device in the hardware environment of Inter (R) i3-2350M CPU 2.7GHz, 4.0GB memory, 1TB hard drive. As shown in Fig.5, ten samples have been selected where small circles represent the ideal corner vertices, the starting point and the end point of the strokes are always considered as corner points. The experimental result of the sample is shown in Fig. 6, and the red real dot represents the extracted Split points.

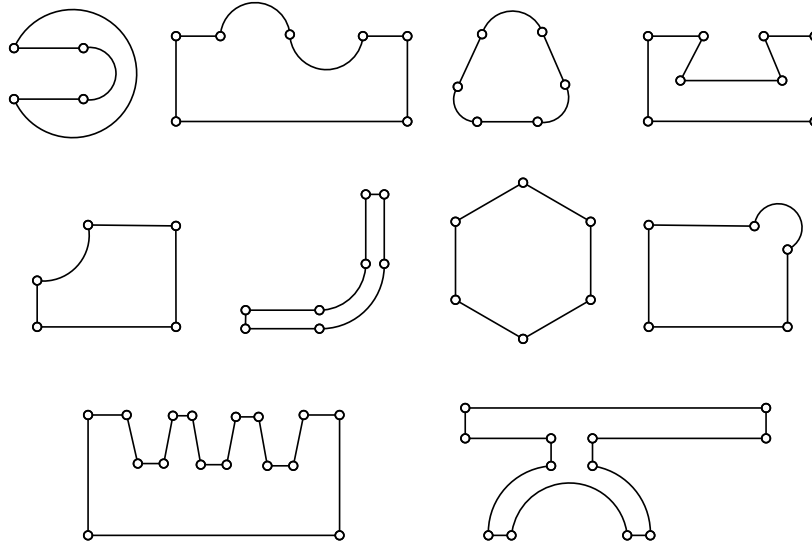


FIGURE 5. Experimental samples.

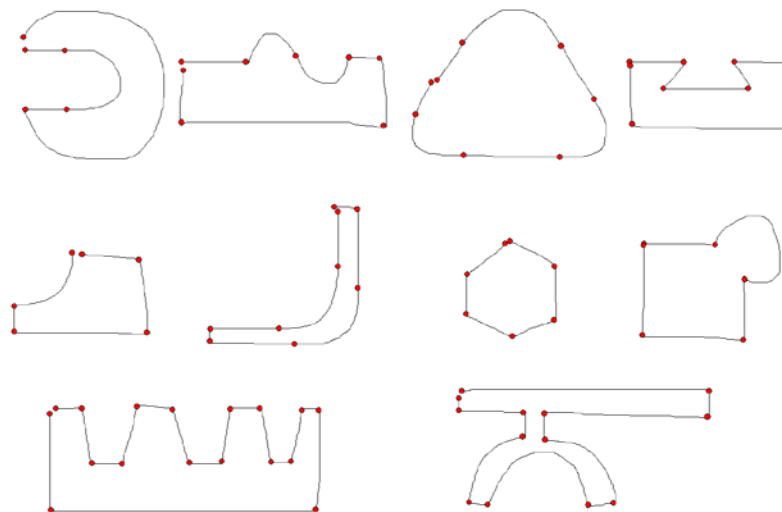


FIGURE 6. The result of experiment.

In order to further analyze the correct rate of the stroke segment method proposed in this paper. First we give some definitions: CN is the number of ideal segmentation points that users intend to express; CC is the number of corner points using our algorithm; FP is the number of false corners

when FN is the number of real corners which are not recognized by our method. Then the recognition accuracy rate CCR can be calculated by the equation:  $CCR = (CC/CN) \times 100\%$ , the misrecognition rate FPR is  $FPR = (FP/(FP + CC)) \times 100\%$ . Second, we collected the sample data from 6 different experimenters, each sample was drawn 3 to 5 times, and finally 220 samples were obtained. Last we choose the ShortStraw algorithm and the ISraw algorithm for comparison, the result can be found in Table 1.

**TABLE 1.** Three algorithm comparing.

	<b>ShortStraw Algorithm</b>	<b>ISraw Algorithm</b>	<b>Algorithm in This Paper</b>
FP	497	137	58
FN	223	253	106
CC	1457	1427	1541
CN	1680	1680	1680
CCR	86.7%	84.9%	91.7%
FPR	25.5%	8.8%	3.6%

As we see from Table 1, due to the existence of a certain proportion of tangent points, the correct rate of ShortStraw algorithm and ISraw algorithm have decreased in some degree comparing to the literature mentioned. At the same time, ShortStraw algorithm has is a high error recognition rate when the algorithm proposed in this paper has a high recognition accuracy rate, and the pseudo-split points are relatively small. The leakage points are mainly due to the fact that the user draws straight lines and arcs too casual, resulting in systematic misjudgment and losing the split points, which also makes the request on drawers.

#### 4. CONCLUSION

This paper presents an effective segmentation method for hand-drawn sketches. First, the stroke is sampled, de-noised and other pre-processing to obtain the re-sampling point sequence; second the approximate curvature of each point is calculated and the discrete curvature threshold is set to get the initial corner point set; If the fitting error of basic primitives between any pair of corner points exceeds the given threshold, the tangent point is extracted by using the vector outer product or the parametric polynomial approximation method to get the final corner set. Finally the sketch segmentation is realized. Experiments show that the algorithm proposed in this paper can be used for segmentation of hand-drawn sketch effectively.

#### REFERENCES

- [1].Company P, Contero M, Varley P. Computer-aided sketching as a tool to promote innovation in the new product development process. *Computers in Industry*, 2009. 60(8): p. 592-603.
- [2].Paulson B, Hammond T. Paleosketch: accurate primitive sketch recognition and beautification: Proceedings of the 13th international conference on Intelligent user interfaces, 2008[C]. ACM.
- [3].Lee W, Kara L B, Stahovich T F. An efficient graph-based recognizer for hand-drawn symbols [J]. *Computers & Graphics*, 2007, 31(4):554-567.
- [4].Kim D H, Kim M. A curvature estimation for pen input segmentation in sketch-based modeling [J]. *Computer-Aided Design*, 2006, 38(3):238-248.
- [5].Wolin A, Eoff B, Hammond T. ShortStraw: A Simple and Effective Corner Finder for Polyline[C]//SBM. 2008: 33-40.
- [6].Xiong Y, LaViola Jr J J. A shortstraw-based algorithm for corner finding in sketch-based interfaces [J]. *Computers & Graphics*, 2010,34(5):513-527.
- [7].Pu J, Gur D. Automated freehand sketch segmentation using radial basis functions [J].

Computer-Aided Design, 2009, 41(12):857-864.

- [8].Albert F, Fernández-Pacheco D G, Aleixos N. New method to find corner and tangent vertices in sketches using parametric cubic curves approximation [J]. Pattern Recognition, 2013, 46 (5):1433-1448.
- [9].Herold J, Stahovich T F. A machine learning approach to automatic stroke segmentation [J]. Computers & Graphics, 2014,38:357-364.