IEICE TRANS. INF. & SYST., VOL.E95–D, NO.11 NOVEMBER 2012

PAPER

A Line Smoothing Method of Hand-Drawn Strokes Using Adaptive Moving Average for Illustration Tracing Tasks

Hotaka KAWASE[†], ***Nonmember***, **Mikio SHINYA**[††a)], ***Member***, ***and*** **Michio SHIRAISHI**[††], ***Nonmember***

SUMMARY There are many web sites where net users can post and distribute their illustration images. A typical way to draw a digital illustration is first to draw rough lines on a paper and then to trace the lines on a graphics-tablet by hand. The input lines usually contain fluctuation due to hand-drawing, which limits the quality of illustration. Therefore, it is important to remove the fluctuation and to smooth the lines while maintaining sharp features such as corners. Although naive applications of moving average filters can smooth input lines, they may cause over-smoothing artifacts in which sharp features are lost by the filtering. This paper describes an improved line smoothing method using adaptive moving averages, which smoothes input lines while keeping high curvature points. The proposed method evaluates curvatures of input lines and adaptively controls the filter-size to reduce the over-smoothing artifacts. Experiments demonstrated advantages of the proposed method over the previous method in terms of achieving smoothing effect while still preserving sharp feature preservation.

key words: *moving average, illustration tracing, smoothing hand-drawn lines, computer graphics*

1. Introduction

There are many web sites where net users can post and distribute their digital illustration images and exchange their comments and evaluations, which have greatly enlarged non-professional digital illustration communities. Although digital illustration can be created fully on-line by using paint systems, many users still prefer to draw rough lines on a paper first*. After scanning images, they trace the lines on a graphics-tablet by hand, and finally complete their colored illustration on paint systems. The input lines usually contain fluctuation due to hand-drawing, which limits the quality of illustration. In particular, it is generally a hard task for non-professional i llustrators to draw mildly curved smooth lines. Applying low-pass filters can smooth input lines. Morimoto proposed a smoothing method by using moving averages [1]. Unfortunately, however, the method tends to cause over-smoothing artifacts, rounding out high curvature parts. Since sharp features such as corners are important components in illustration, missing these features may fatally damage the illustration. It is desirable to smooth drawn lines while keeping these features.

This paper describes an improved line smoothing method using adaptive moving averages, which smoothes input lines while keeping high curvature feature points. In Morimoto's method [1], a fixed number of input points are averaged over. In our method, on the other hand, we control the filtering size based on the estimated curvature. The curvature is first estimated by least-square-fitting to the input points. The filtering can be progressively performed by updating curvature estimation by using the previous filtering result. The optimal filter sizes at given curvature values were empirically determined through measurements. Experiments demonstrated advantages of the proposed method over the previous method in terms of achieving smoothing effects while still preserving sharp features.

2. Related Work

Although image processing techniques such as filtering and segmentation have been actively developed and applied to painting systems, only a few papers have addressed issues concerning to reshaping strokes for illustration tracing tasks, as far as the authors know.

In some sketched diagrams, such as figures used in physics and mathematics classes, it is possible to assume geometric constraints such as parallelism, symmetry and perpendicularity in those figures. Igarashi applied such constraints to reshape input hand-drawn strokes for beautification [2]. When drawings can be assumed to be composed of a set of simple primitives, such as lines, ellipses and circles, it is possible to apply stroke recognition techniques to classify input lines into curve segments each of which is represented by the primitives [3]. In general situations, however, these geometric assumptions are not always realistic. For example, Anime-like illustrations are usually composed of free-form curves and it is difficult to assume simple geometric constraints such as synmetry and lineality in those illustrations.

For the purpose of smoothing general hand-drawn lines in illustration, two approaches have been proposed, these are, the spline-based approach and the filter-based approach. In the spline-based approach, spline curves are fitted to the input points and the resulting smooth curves are used as the modified lines [4], [5]. Low-pass filters naturally smooth input signals and can be applied to this problem, as well. Morimoto [1] proposed a smoothing method based on moving

Manuscript received March 19, 2012.
Manuscript revised June 12, 2012.
†The author is with the Mitsubishi UFJ Information Technology, Tokyo, 140–0002 Japan.
††The authors are with Toho University, Funabashi-shi, 274–8510 Japan.
a) E-mail: shinya@is.sci.toho-u.ac.jp
DOI: 10.1587/transinf.E95.D.2704

*We asked 12 amateur illustrators about their illustrating style and found a half of them prefer to trace scanned images.

average filters. This method calculates averages of a fixed number of input points and uses them as the modified lines. These two approaches have different features and useful to different types of applications because of their different output data structures. The filter-based methods are usually simple to implement, result in dot raster images, and are suitable to paint-type applications. The spline-based methods, on the other hand, are more complicated, but result in spline curves and are suitable to draw-type applications.

The both methods smooth input lines, but may cause over-smoothing artifacts, rounding out high curvature parts. Since sharp features such as corners are important components in illustration, missing these features is an annoying side-effect. In order to prevent such artifacts, it is necessary to introduce some adaptive mechanism that controls smoothing process into the both approaches.

Since, in the illustration tracing tasks, the smoothed lines are usually handled by paint systems, we selected the moving average approach and introduced an adaptive control scheme to the moving average process. The next section analyses the moving average method and introduces key ideas to overcome the over-smoothing problem.

3. Moving Average Method

The moving average is a simple low-pass filtering process. Let P_i be a 2D vector representating the x- and y- coordinates of the i-th input point. The modified point, M_i, is calculated by[†]

$$M_i = \sum_{-N/2}^{N/2} P_{i+j}/N, \tag{1}$$

where N denotes the filter size. This smoothes fluctuations involved in the input points (Fig. 1).

Next, let us consider the over-smoothing artifacts in high curvature regions. In the example shown in Fig. 2 (a), the curvature at the input point P_i is high. Since all of its neighboring points, P_{i+j} and P_{i-j}, are located in the left-hand side of P_i, the modified point M_i shifts to the left side, and the modified line has lower curvature near M_i. This is the over-smoothing artifact. A simple way to ease this problem is to decrease the filter size at high curvature points.

Artifacts may occur even at low curvature points. In the example of Fig. 2 (b), although the curvature at P_i is low, a sharp corner P_{i+2} exists in the filtering interval, which makes curvature at M_i higher. From these analyses, it is considered that the following strategy can ease the problems:

- to decrease the filter size at high curvature points,
- to limit the filtering interval not to exceed a high curvature point.

In the next section, a new smoothing method is proposed based on this consideration.

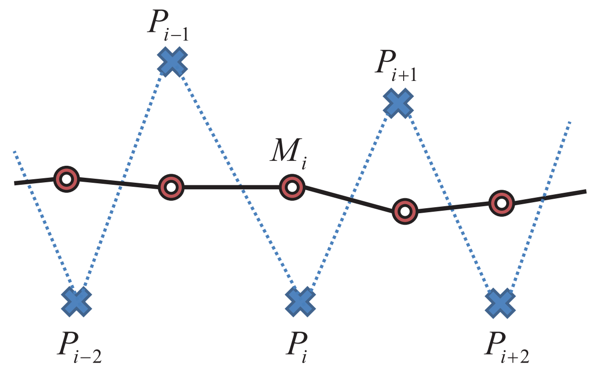
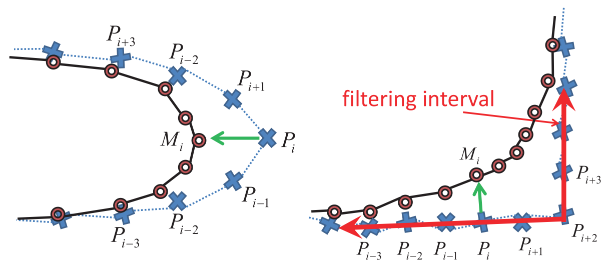


Fig. 1 Moving average.



(a) (b)

Fig. 2 Over-smoothing artifacts.

4. Adaptive Moving Average Method

Based on the discussion in the previous section, we propose an improved moving average method with adaptive filter-size control. We first empirically determined the optimal filter-size for each curvature value from measurements. Then, a curvature estimation method is described, and finally, the improved algorithm is proposed.

4.1 Optimal Filter-Size

The optimal filter-size was empirically determined from measurements. We first collected data samples of traced lines, and applied the moving average filter with several filter-sizes. From analyses of the filter results, we determined the optimal filter-size.

4.1.1 Data Collection

An ellipse with 320 dots by 120 dots was displayed as a target curve on a screen, and four subjects were asked to trace it on an A6 size graphics-tablet. Each subject traced three times. Each input point P_i is associated with the closest point T_i on the target curve (Fig. 3), and the target curvature at P_i, $1/R(P_i)$, is defined as the curvature of the target curve

[†]In order to process the starting point, P_0, and the end point, P_m, we set $P_i = P_0$ for $i < 0$ and $P_i = P_m$ for $i > m$, and the averages are calculated from $i = -N/2$ to $m + N/2$.

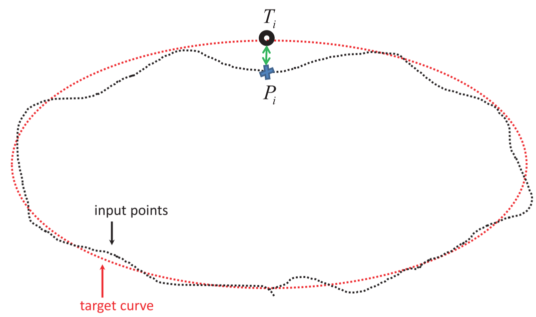


Fig. 3 Traced target curve.

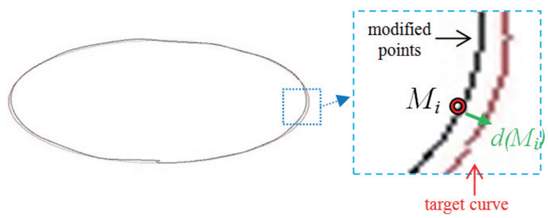


Fig. 4 Evaluation of filtering results.

at T_i. The input points were grouped according to their target curvature values.

We first thought that the pen speed might be an important factor, but preliminary experiment suggested that it was not, and thus, we decided not to control the speed in the measurements.

4.1.2 Evaluation of Filtering Results

To each collected input lines, we applied the moving average filter with a fixed filter-size, ranging from 5 to 100 points. For each modified point M_i, we calculated the minimum distance to the target curve, $d(M_i)$ (Fig. 4). Since large $d(M_i)$ values indicate undesired results, we adopted $d(Mi)$ as the evaluation of the filter at M_i.

4.1.3 Optimal Filter-Size

According to the target curvature of the corresponding input points, the distance $d(M_i)$ was summed up. Figure 5 shows a result for input points whose target curvature is higher than 1/10 dots. In this example, the least sum was obtained at filter-size=20, and the optimal filter-size is determined to be 20 points.

Figure 6 shows the averages and variations of the measured optimal filter-sizes for each curvature range among the subjects. Although there were some variations among subjects, overall features were similar: when the line curvature is large, the optimal filter-size becomes small.

We also examined the influence of the tablet size, which scales the physical size of the pen movement. Figure 7 shows the measured optimal filter-size for each curvature range with respect to graphics-tablet sizes (A6, A5 and A4). As shown in the figure, it was found that the influence

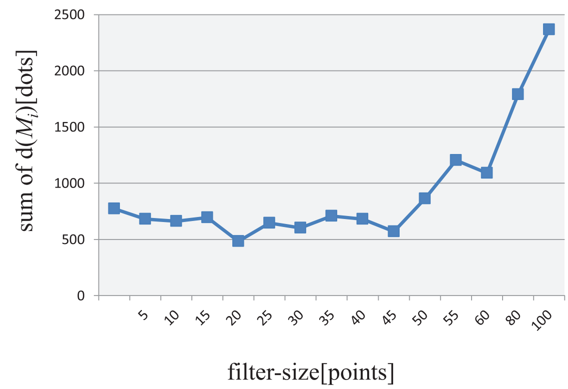


Fig. 5 An example of filter evaluation (target curvature > 1/10 dots).

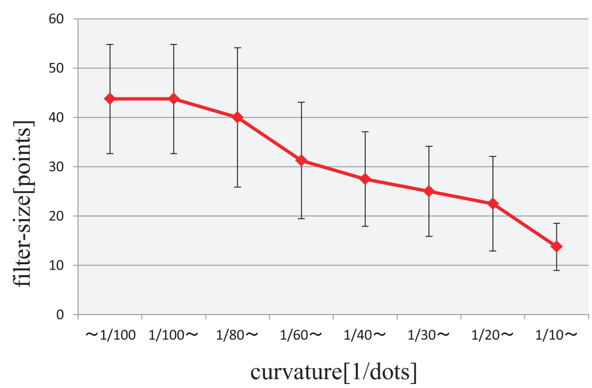


Fig. 6 The optimal folter-size.

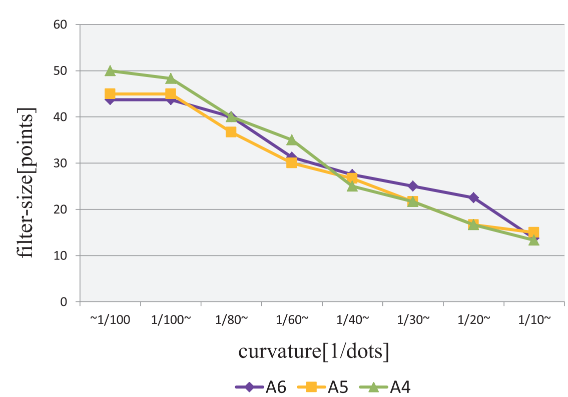


Fig. 7 The optimal filter-size and tablet size.

of graphics-tablet size is very small.

Based on these observations, we decided to simply use the average values as the optimal filter-size, and saved them as a table, called the optimal filter-size table, so as to be referred to during the smoothing process.

4.2 Curvature Estimation

From given points (x_i, y_i), the curvature q can be estimated by a least-square scheme, in the following way.

1. A straight line $L(x) = ax + b$ is fitted so as to minimize

$\sum_i |L(x_i) - y_i|^2|$ (Fig. 8 (a)),

2. Set a new coordinate using the direction of line L as the x'-direction and the center points as the origin. Transform (x_i, y_i) to (x'_i, y'_i) (Fig. 8 (b)).
3. A quadric curve $Q(x') = (q/2)x'^2$ is fitted so as to minimize $\sum_i |Q(x'_i) - y'_i|^2$ (Fig. 8 (c)).

The obtained q value is regarded as the curvature value at the center point. As the given points, we use the input points at first, and then the modified points can be used to refine the estimation when necessary.

4.3 Proposed Method

Figure 9 outlines the procedure of the proposed method. From the input point data $\{P_j\}$, the curvature q_i is estimated. Using the curvature q_i, the optimal filter-size N_{opt} is determined by referring to the optimal filter-size table.

According to the discussion in Sect. 3, we check the curvature values of the input points in the filter interval. If we find a curvature exceeding a given threshold value, the interval is truncated so that it does not contain a high curvature point.

Applying the moving average filter with the determined interval results in the modified point M_i. When users are not satisfied with the results, they can enforce the system to iteratively re-evaluate the curvatures using the calculated modified points, which may reduce curvature estimation errors.

5. Experiments

Two illustration images (CHERRY-BLOSSOM and RABBIT) were traced by three subjects. The obtained input points were smoothed by the previous moving average method (MA) [1] and the proposed method (AMA), and we investigated how effectively these methods smoothed the input lines while maintaining the sharp features. The filter size of the moving average method was set to 40 points, which we found to provide visually good results. In this experiment, the never-exceed curvature threshold value was set to 20 dots.

Figure 10 (a) shows an example of the input lines traced over the CHERRY-BLOSSOM image, where high curvature points (> 1/20 dots) were plotted by green points. Visually important corners were manually selected and marked by blue circles. As shown in the figure, the input lines contain a lot of high curvature points, indicating hand-drawing fluctuations. Figure 10 (b), (c) and (d) show the filtered results by the moving average method (MA), by the proposed method with curvature estimation from the input lines (AMA1), and by the proposed method with four times repeated estimation (AMA4), respectively. As shown in the figures, all the filters succeeded in smoothing the input lines, reducing the number of the high curvature points. However, the MA method over-smoothed the feature points, as shown in the area indicated by a rectangle of red dashed-lines in the figure. The proposed method, on the other hand, maintained sharpness of most of the feature corners.

Figure 11 shows an example of the RABBIT illustra-

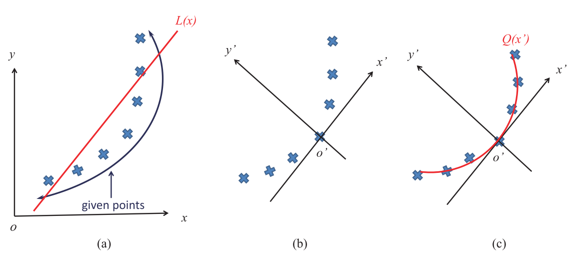


Fig. 8 Curvature estimation by least-square-fitting.

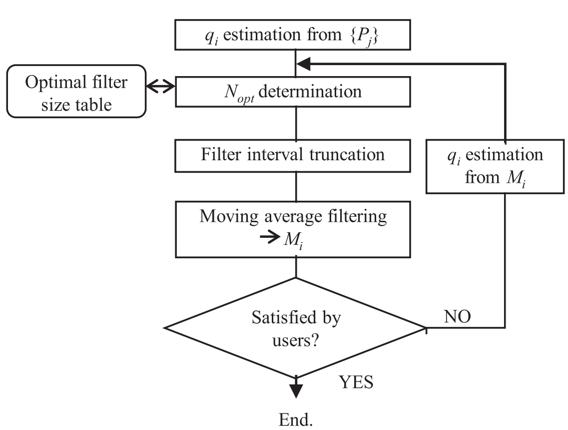


Fig. 9 The procedure of the proposed method.

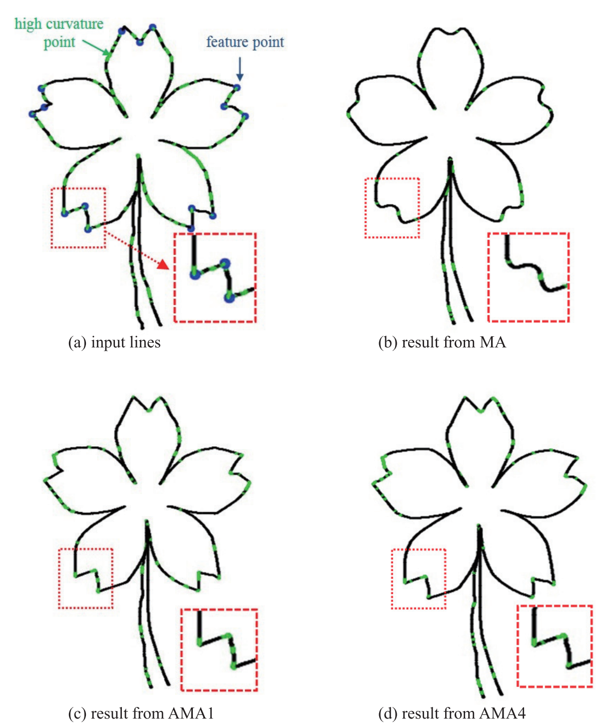
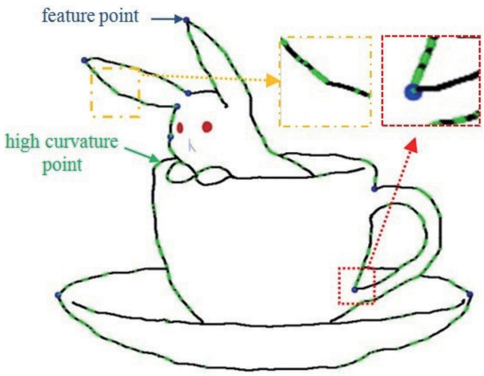
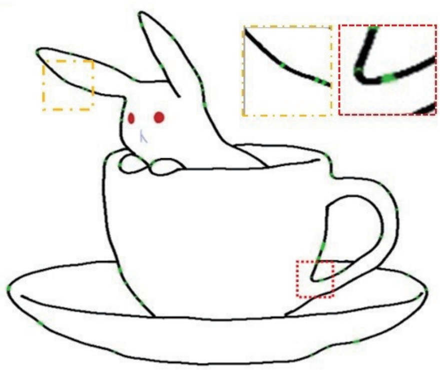


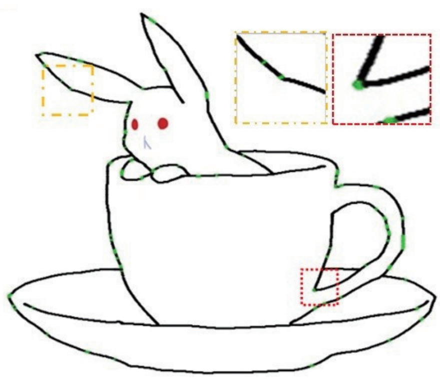
Fig. 10 CHERRY-BLOSSOM illustration.



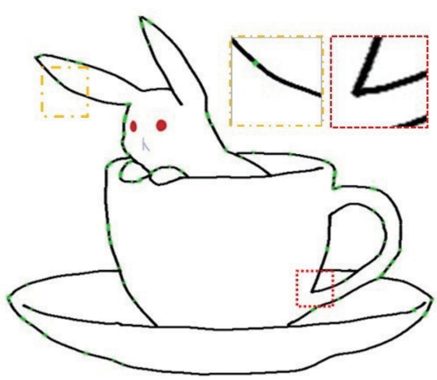
(a) input lines



(b) result from MA



(c) result from AMA1



(d) result from AMA4

Fig. 11 RABBIT illustration.

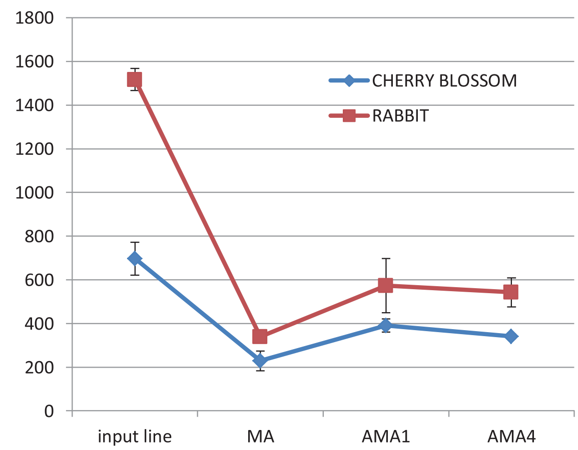


Fig. 12 The number of high curvature points (1>20 dots), averaged over three subjects.

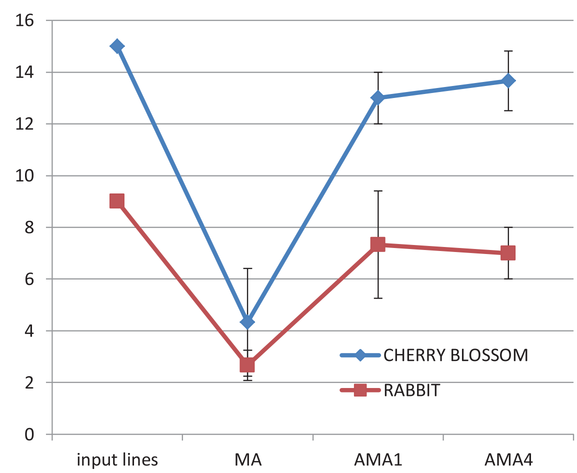


Fig. 13 The number of the preserved feature points, averaged over three subjects.

tion. Tendencies similar to the CHERRY-BLOSSOM example can be observed here: the proposed method well smoothed the input lines while keeping sharp features. Note that AMA4 provided a slightly better result than AMA1 near the right ear part, marked by a yellow dashed-line rectangle in the figure, in this example.

To quantitatively compare the results of the proposed methods (AMA1 and AMA4) and the previous method (MA), we counted the number of high curvature points (Fig. 12). As shown in the figure, all of the three methods considerably reduced the number of high curvature points, indicating effective smoothing results. We also counted the number of the feature points preserved in the filtering results (Fig. 13). Here, we judged a feature point P_i was preserved when there is a high curvature point (i.e., >1/20 dots) in the neighborhood (±2 dots) of the corresponding modified point M_i. As seen in the figure, the previous method (MA) missed most of the feature points. The proposed method (AMA1 and AMA4), on the other hand, succeeded in preserving most of these features, keeping high curvature corners. The

Table 1 Increased number of preserved feature points from the previous method (MA) to the proposed method (AMA1). The average among the three subjects, the standard deviation, t-value and p-value are indicated for the two illustration images.

	Ave. (AMA1-MA)	SD.	T	p
CHERRY-BLOSSOM	8.7	1.25	9.8	< 0.01
RABBIT	4.7	2.1	3.2	< 0.05

Table 2 Processing time per point, measured on a Windows 7 64 bit PC with an Intel Core i7 950 at 3.07 GHz and 6 GB main memory.

	Perevious (MA) ms	Proposed (AMA1) ms
CHERRY-BLOSSOM	2.8	53
RABBIT	4.7	39

variance among the subjects were small. Table 1 shows the increased number of preserved feature points from the previous method to the proposed method, and t-tests indicated significant increases between them.

The proposed method shown in Fig. 9 is simple enough to achieve real-time operation even on current note-PC environments. Table 2 shows the measured processing time per point with the two illustrations. Although processing time incerases from the previous method mainly due to the curvature estimation, the proposed method still allows interactive operations.

6. Conclusions

This paper described an improved line smoothing method using adaptive moving averages, which smoothes input lines while keeping high curvature points. The proposed method evaluates curvatures of input lines and adaptively controls the filter-size to reduce the over-smoothing artifacts. Optimal filter size was empirically determined from measurements, and the curvatures were estimated by a least square scheme. Experiments demonstrated advantages of the proposed method over the previous methods in terms of achieving smoothing effects while still preserving sharp features.

Filter sizes in moving average roughly corresponds to control-point intervals in spline curves. Future work includes applications of the adaptive control to the spline-based smoothing methods to improve their smoothing performance.

References

[1] D. Morimoto and M. Nawate, "The proposal of a paint tool with a blurring compensation function," The Institute of Electronics, Information and Communication Engineers, vol.102, pp.53–58, 2003.
[2] T. Igarashi, S. Matsuoka, S. Kawachiya, and H. Tanaka, "Interactive beautification: A technique for rapid geometric design," Proc. UIST, pp.105–114, 1997.
[3] B. Paulson and T. Hammond, "PaleoSketch: Accurate primitive sketch recognition and beautification," Proc. IUI'08, pp.1–10, 2008.
[4] R. Pusch, F. Samavati, A. Nasri, and B. Wyvill, "Improving the sketch-based interface: Forming curves from many small strokes," The Visual Computer, vol.23, no.9-11, pp.955–962, 2007.
[5] K. Matsuda and K. Kondo, "Interactive sequential drawing for freehand sketch inputting," Trans. IPSJ, vol.40, no.2, pp.594–601, 1999.

Hotaka Kawase received the B.S. and M.S. degrees in Information Science from Toho University in 2010 and 2012, respectively. He currently works with Mitsubishi UFJ Information Technology, Ltd.

Mikio Shinya is currently a Professor at Department of Information Science, Toho University. He received a BSc in 1979, an MS in 1981, and a PhD in 1990 from Waseda University. He joined NTT Laboratories in 1981, and moved to Toho University in 2001. He was a visiting scientist at the University of Toronto in 1988–1989. His research interests include computer graphics and visual science.

Michio Shiraishi was born in 1974. He received his Ph.D. degree from The University of Tokyo in 2003. He has been a lecturer of Toho University since 2005. His current research interests are computer graphics and web application systems.