Tidy writing tool using time information

Shinji Sakai, Kazuo Misue, Jiro Tanaka

Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba
1-1-1 Tennoudai, Tsukuba, Ibaraki, 305-8573, Japan
{sakai, misue, jiro}@iplab.cs.tsukuba.ac.jp

Abstract. Electronic whiteboards are capable of preserving and reusing writings on blackboards (or whiteboards) at classes. This means that many more people than before can see the writings. Making these writings tidy is a desired feature. This paper describes a writing tool on the electronic whiteboards. The tool uses time information to treat strokes as two levels of chunks: the character level and the sentence level. The tool makes the writings tidy by using the chunks in the character level and replays efficiently the writing processes by using the chunks in the sentence level.

1 Background

Recently, the opportunity to see large screen devices such as plasma displays and projectors has increased. The classes and presentations using these large screen devices have increased too. The literatures prepared beforehand are used in most of those classes and presentations. It is difficult to change the contents and orders of the lectures and presentations according to the place. Writing on a blackboard in the place of the classes is a very flexible way which allows changes of contents and orders of the lectures and the presentations. Most writings on blackboards have been erased, by the end of the classes. In a word, writing on a blackboard has been used and then thrown away. However, the appearance of electronic whiteboards like Tivoli[1] has changed the situation. Preserving such writings electronically became possible and therefore reusing these writings became possible as well.

2 Writing system on the electronic whiteboard

In this paper, writing on the blackboard in classes and presentations with the purpose of being shown to on audience is called "blackboard writing." By the appearance of electronic whiteboards, it seems that a blackboard writing is seen by many people more than before, and reused over and over again.

The tidy blackboard writing is desired because many people might see a blackboard writing. Because blackboard writings are written by hand, they can be messy;

alteration of the strokes' shape due to hand swinging, misalignment of string baselines, and inaccurate figures may occur. Such messiness needs to be resolved.

Besides, writing flow is important as well as contents of blackboard writings. It is difficult to figure out the flow only by seeing its final form. When blackboard writings are reused it is needed to replay not only contents of writings but also the flow.

We aim to develop a writing tool that preserves writings in tidy form with flows, and replays them. The tool has two features. First, the tool makes writings tidy in real time. Making writings tidy later increases the amount of the work. So, the tool makes writings tidy without increasing the amount of the user's work by processing in real time. Second, the tool replays writings flow. Writing flows are replayed by redrawing strokes in writing along the time when writings are reused.

3 Using time information on the writing tool

Our tool treats strokes as chunks by the unit of one character or one sentence. In replaying strokes by each stroke, process of writing a character is also replayed when writing is reused. However, such replaying gives the user a halting impression. So, the tool treats strokes as chunks, and replays writing flow efficiently by redrawing strokes by each chunk. We assume that a chunk is a unit of sentence (formula). In addition, the tool uses these chunks for making writings tidy, too. When aligning the string baselines, these chunks are used to move strokes in each character in order to keep the shape of characters that consist of two or more strokes. At the higher level, in sentences and formulas, the tool treats characters as chunks. At the lower level, in characters, the tool treats strokes as chunks.

The tool uses the no-input time for making chunks of strokes, that is the time when no input takes place. Because one character is usually written continuously, we believe that the time between writing the strokes that compose it is relatively short. Our tool measures the time interval between the moment a stroke has been finished and the moment the next stroke has started to be drawn. A chunk is recognized when this time exceeds a threshold previously set. The tool sets two thresholds in order to treat strokes as two levels of chunks: the character level and the sentence level.

The tool starts timing when the drawing of a certain stroke A has been finished. If the drawing of the following stroke B is started before the time exceeds the set threshold, the timing is stopped. The tool starts a new timing when the drawing of B is finished. When the no-input time exceeds the character level threshold, strokes up to A are recognized as a character level chunk. At this time, the tool informs the user that this is a chunk by changing the color of the strokes involved. (Fig.3-1)

Our tool calculates the distance between the character level chunk and the baseline and moves the strokes so that they are aligned to the baseline. When the writing is reused, the system replays strokes by each sentence level chunk.

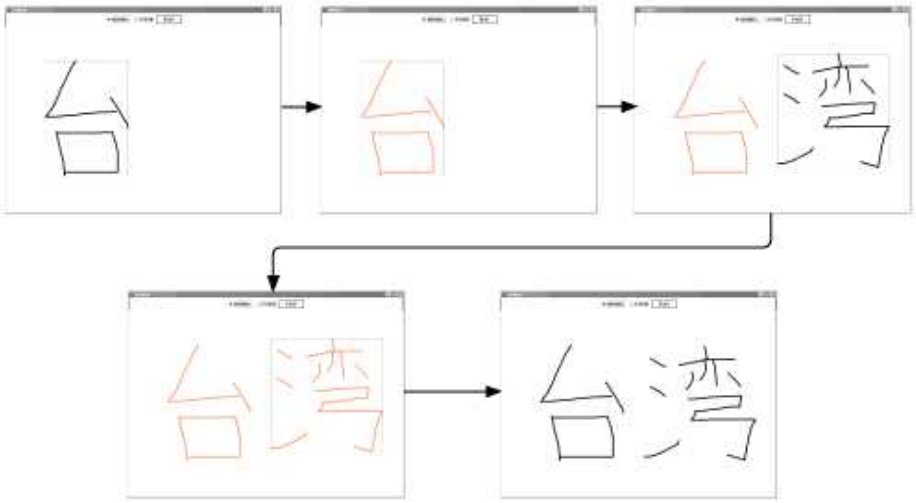

Fig.3-1. Chunking flow

4 Blackboard writing tool “twilight”

We developed a blackboard writing tool "twilight" based on the idea described in Section 3. Twilight makes writings tidy in real time by reducing alteration of the strokes’ shape due to hand swinging, recognition and forming hand-written figures, and aligning string baselines using chunks of strokes. When writing is reused, twilight replays writing flow efficiently by redrawing strokes by each sentence level chunk. As described in Section 3, twilight has the feature to treat strokes as two levels of chunks: the character level and the sentence level. The character level chunk is set of strokes. The sentence level chunk is set of character level chunks.

4.1 Start screen

Fig.4-1 is the start screen of twilight. 1 is the writing space where the user writes strokes. 2 is the radio button for switching the writing modes. 3 is the button for beginning the replaying writings.

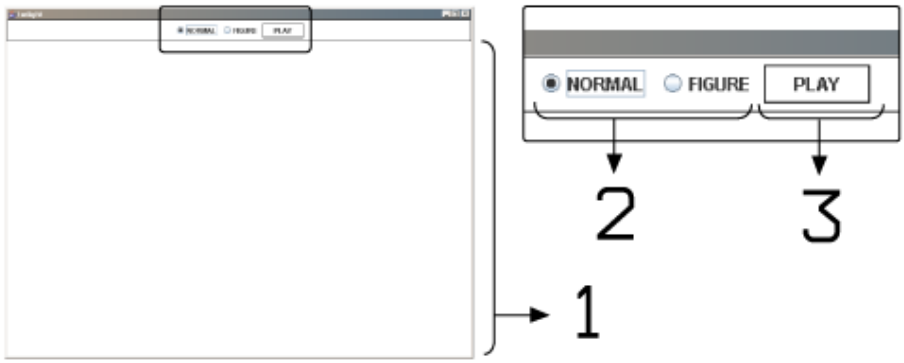

Fig.4-1. Start screen of twilight

4.2 Writing mode

There are two writing modes in twilight. One mode is the free stroke mode. In this mode, the user can write characters and free strokes. In this mode, alteration of the strokes' shape due to hand swinging is reduced and string baselines are aligned. The other mode is the stroke recognition mode. In this mode, the user can draw a straight line, a rectangle, and an oval accurately. Figures drawn by the user are automatically recognized and formed. Default mode is the free stroke mode. These modes are switched by the radio button on the upper of screen.

4.3 Reducing alteration of the strokes' shape due to hand swinging (in the free stroke mode)

Blackboard writing is different from writing on paper. It is writing on a vertical surface. In this case, the user can't have a point of support for writing alike writing on paper. Therefore, alteration of the strokes' shape due to hand swinging might occur. Twilight reduces such alteration by using the Bezier approximation. In the free stroke mode, twilight calculates the formula of the third Bezier curve from points of a stroke when the stroke is written. Then, twilight changes points of the stroke based on its formula. Fig.4-2 is the appearance of actual processing.

Fig.4-2. Reducing the alteration of a stroke's shape due to hand swinging

4.4 Alignment of baselines of strings (in the free stroke mode)

The entire scale of blackboard writing is larger than that of writing on paper. Therefore, it is difficult for the writer to figure out the whole image of the writing and misalignment of string baselines might occur as a result. Twilight automatically aligns string baselines. First, whether it is columnar writing or horizontal writing is recognized from the positional relation of first two characters in the string. As for horizontal writing, because the user naturally tries to align the characters horizontally, we thought that the vertical distance is shorter than the horizontal distance. The direction of the string is recognized by checking which distance in the horizontal direction and the vertical direction is longer. The string baseline is aligned according to the position of the first character.

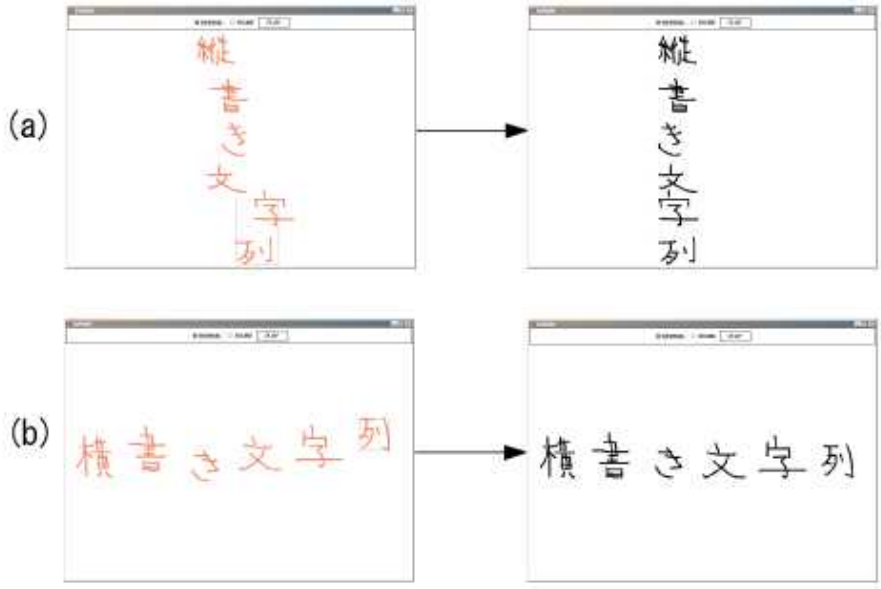

Fig.4-3. Alignment of baselines

4.5 Recognition and forming of figures (in the stroke recognition mode)

Figures are often used in blackboard writing. When figures are used, some accuracy is desired. However, it is difficult to draw figures accurately by the hand. Twilight recognizes hand-written figures and forms to the most likely figures automatically. Figures which are recognized and formed are the straight line, the oval, and the rectangle because we thought that these figures are often used in blackboard writing. As for the oval and the rectangle, only the part where enclosed in the stroke is recognized and formed.

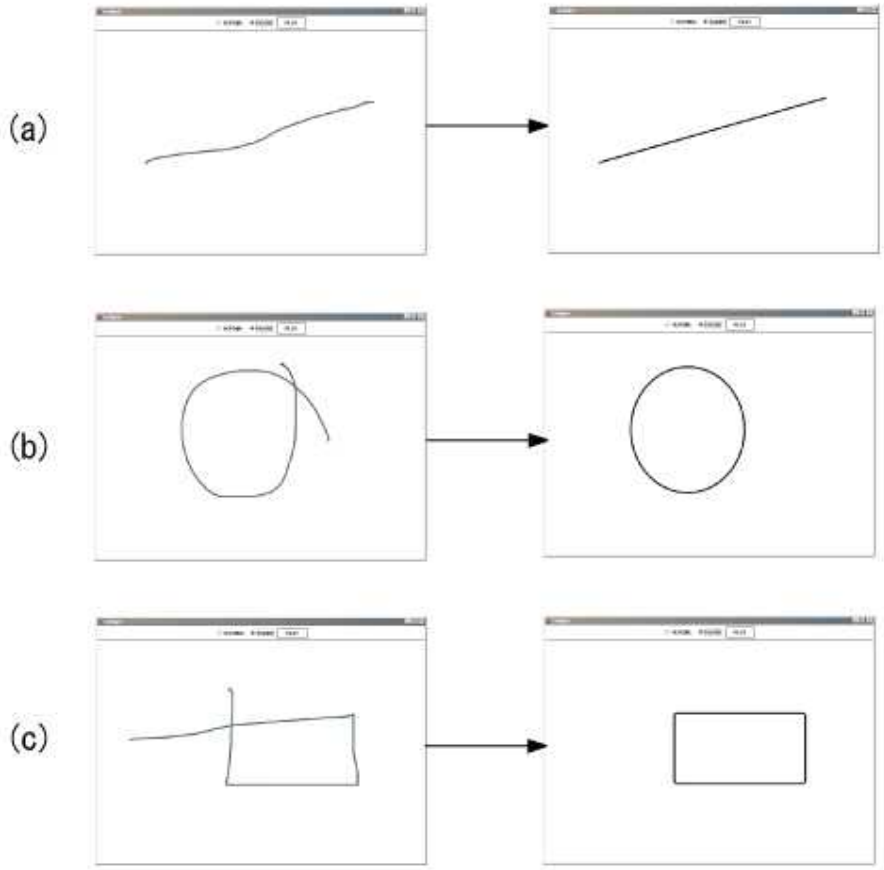

Fig.4-4. Recognition and forming of figures:
(a) is the straight line, (b) is the oval, (c) is the rectangle.

4.6 Replaying writing

In twilight, the user can see writing completed previously along the writing flow. The user can observe not only the contents of writing, but also the writing flow by watching the strokes being drawn in real time. However, this takes the same amount of time as the initial writing. Moreover, these are time intervals without any input. In a word, replaying all of writing in real time is not efficient. In twilight, strokes are redrawn at each sentence level chunk in order to replay writing efficiently.

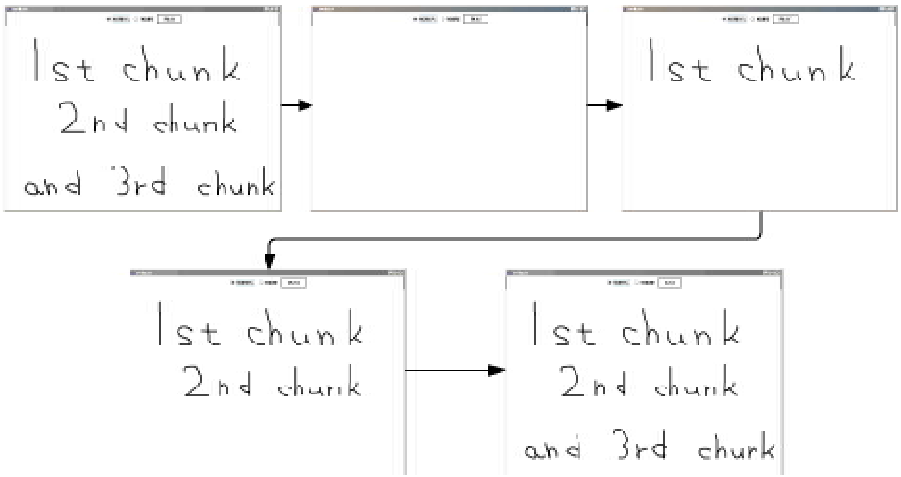

Fig.4-5. Replaying writing at each sentence level chunk

5 Evaluation

We evaluated twilight from two points of view. One is the utility of making writings tidy in real time. Another is the utility of replaying writings at each sentence level chunk. We conducted the evaluation experiment on 8 examinees. The procedure of the experiment is as follows.

For writing

Exp.1 Examinees produce a blackboard writing that consist of 4 lines, 50 characters in Japanese, without the function of making the writing tidy in real time: reducing losing shapes of a stroke due to hand swinging, recognition and forming of figures, and alignment of baselines of strings.

Exp.2 They produce the same blackboard writing as Exp.1 using the function of making the writing tidy in real time.

For playing

Exp.3 Examinees see the blackboard writing that consist of 6 items in Japanese, and guess the order of items.

Exp.4 Examinees see the same blackboard writing as Exp.3 replayed, and guess the order of items.

We evaluated the utility of making writings tidy in real time by the examinees' subjectivity. We evaluated the utility of replaying writings at each sentence level chunk by the examinees' subjectivity and the correct guess rate of the order of items. The correct guess rate is the rate of the correct order set of 15 sets that is the number of combination that takes 2 items from 6 items. The evaluation item and the result are as follows.

For writing

1.1 Utility of reducing alteration of the strokes' shape due to hand swinging
1.2 Utility of recognition and forming of figures
1.3 Utility of alignment of baselines of strings
1.4 Utility of treating strokes as chunks
1.5 Which writing is the one that the writer wants to show to the audience?

For playing

2.1 In which writing is it easier to understand the writing flow?
2.2 Which writing is conspicuous?
2.3 Whether the replaying method is appropriate or not.

(As for 1.5, 2.1, and 2.2 the examinees select twilight(1) or not(0). As for others, they evaluate each item on a scale from 1 to 5 where 5 is assumed to be a high evaluation value.)

Table.5-1. Evaluation result of making writing

Examinee	1.1	1.2	1.3	1.4	1.5
A	3	4	1	4	0
B	3	5	5	5	1
C	4	2	5	3	0
D	4	4	3	2	1
E	2	5	4	3	1
F	4	5	5	3	1
G	5	4	4	5	1
H	4	4	5	3	1

Table.5-2. Evaluation result of replaying writing

Examinee	the correct guess rate (Exp.3 → Exp.4)	2.1	2.2	2.3
A	66% → 100%	1	1	4
B	80% → 100%	1	-	4
C	80% → 100%	1	-	3
D	60% → 100%	1	1	3
E	80% → 100%	1	1	2
F	80% → 100%	1	1	2
G	80% → 100%	1	-	3
H	80% → 100%	1	1	4

We received some comments from the examinees through the evaluation experiment. Firstly, there were two comments about alignment of baselines of strings. One examinee pointed that sometimes a sentence has overlapped with other sentence partly when baselines of strings have been aligned. One other examinee pointed that baseline alignment is not convenient because even when the writer wants to write

one long sentence in two lines, the string baseline is aligned according to position of the first character. We think that it is necessary to arrange stroke positions in consideration of the entire positional relation in the writing in order to solve sentences overlapping. The issue that it is impossible to write one sentence in two lines indicates the need of arranging stroke positions in consideration of the entire positional relation too. In relation to this, there was a comment that there might be other better information for recognition of the sentence level chunk than time information. It is necessary to improve such issues.

Secondly, more than 2 examinees made a similar comment about replaying writing at each sentence level chunk by the click or button operation. We think that such a method is very useful because it is possible to change the speed of the explanation according to the place when blackboard writings are reused. In addition, we think that it is necessary to improve the redrawing method. For example, fade-in or animation might be useful because audience can easily understand where the next chunk is redrawn.

6 Related work

Handwriting is used in various fields. Tools like SILK [2] allow GUI construction using handwriting. Website design tool DENIM [3] made by using SATIN [4] treats layouts by handwriting. SketchREAD [5] and [6] recognize handwritten circuit chart. Pegasus[7] and Fluid Sketches [8] form handwritten figures. As for handwriting systems on electronic whiteboards, tools like Flatland [9] and MathPad [10], which analyzes hand-written formulas and simulates them, exist.

Pegasus forms hand-drawing strokes based on a geometrical restriction. Moreover, it informs the user the following stroke forecasted. As a result, the user can draw interactively on the system. Though Pegasus focuses on drawing geometrical figures, twilight is for free-hand writing. SATIN is a toolkit which supports developing pen-based applications. SATIN can linearize strokes and merge two or more strokes. Fluid Sketches forecasts shape by using the ordinary differential equation and the spring model every time points of the stroke are added. It gives the user earlier feedback by forming strokes dynamically based on the forecast. Twilight recognizes and forms strokes after the strokes have finished writing. However, when the user wants to enclose a string, (s)he should know earlier whether the desired range is encloses or not. There is often such a scene in blackboard writings. Therefore, we think that dynamic feedback like Fluid Sketches is necessary also for twilight.

7 Conclusion and future work

We developed "twilight" as a writing tool on electronic whiteboards. Twilight is a tool to make and to reuse blackboard writings. When writings are made, twilight

makes writings tidy in real time by reducing alteration of strokes' shape, recognition and forming figures, and aligning string baselines using chunks of strokes. When writings are reused, twilight can replay writing flow efficiently using chunks of strokes which chunked when the writing is made. Twilight treats strokes as chunks by time information. Twilight treats strokes as two levels of the chunks: the character level and the sentence level.

Our future work includes improving the interface of twilight in consideration of comments obtained by the evaluation experiment, and adding the function of arranging stroke positions automatically in consideration of the entire positional relation.

References

1. Elin Ronby Pedersen, Kim MacCall, Thomas P. Moran, and Frank G. Halasz. Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings. In: Proceedings of Conference of Human Factors in Computing Systems (CHI 1993), pp.391-398, 1993.
2. James A. Landay. SILK: sketching interfaces like krazy. In: Proceedings of Conference of Human Factors in Computing Systems (CHI 1996), pp.398-399, 1996.
3. James Lin, Mark W. Newman, Jason I. Hong, and James A. Landay. DENIM: finding a tighter fit between tools and practice for Web site design. In: Proceedings of Conference of Human Factors in Computing Systems (CHI 2000), pp.510-517, 2000.
4. Jason I. Hong and James A. Landay. SATIN: A Toolkit for Informal Ink-based Applications. In: Proceedings of Symposium on User Interface Software and Technology (UIST 2000), pp.63-72, 2000.
5. Christine Alvarado and Randall Davis. SketchREAD: A Multi-Domain Sketch Recognition Engine. In: Proceedings of Symposium on User Interface Software and Technology (UIST 2004), pp.23-32, 2004.
6. Levent Burk Kara and Thomas F. Stahovich. Hierarchical and Parsing and Recognition of Hand-Sketched Diagrams. In: Proceedings of Symposium on User Interface Software and Technology (UIST 2004), pp.13-22, 2004.
7. Takeo Igarashi, Sachiko Kawachiya, Hidehiko Tanaka, and Satoshi Matsuoka. Pegasus: A Drawing System for Rapid Geometric Design. In: Proceedings of Conference of Human Factors in Computing Systems(CHI 1998), pp.24-25, 1998.
8. James Arvo and Kevin Novins. Fluid sketches: continuous recognition and morphing of simple hand-drawn shapes. In: Proceedings of Symposium on User Interface Software and Technology (UIST 2000), pp.73-80, 2000.
9. Elizabeth D. Mynatt, Takeo Igarashi, W. Keith Edwards, and Anthony LaMarca. Flatland: new dimensions in office whiteboards. In: Proceedings of Conference of Human Factors in Computing Systems (CHI 1999), pp.346-353, 1999.
10. Joseph J. LaViola, Jr. and Robert C. Zeleznik. MathPad^2: A System for the Creation and Exploration of Mathematical Sketches. In: Proceedings of ACM Transactions on Graphics (TOG), pp.432-440, 2004.