Formalizing Recognition of Sketching Styles in Human Centered Systems

Danilo Avola, Fernando Ferri, and Patrizia Grifoni

Istituto di Ricerca sulla Popolazione e le Politiche Sociali - Consiglio Nazionale delle Ricerche, Via Nizza 128, 00198 Rome, Italy
{danilo.avola,fernando.ferri,patrizia.grifoni}@irpps.cnr.it

Abstract. Sketch-based interaction enables users' simple communication and it is used to represent concepts and commands in human centered systems. This communication approach can be used in different contexts with different devices. The ink style through which the user performs a sketch is a critical component in the recognition and interpretation processes. In particular, the different users' styles adopted to perform the sketch can introduce over-tracing and/or cross-hatching phenomena that are respectively represented in the sketch like bold style or dashed style. The paper provides an approach to formalize the recognition of the different ink styles (bold, solid and dashed) performed by the user during his/her sketch activity.

Keywords: Human centered systems, sketch-based interaction, ink styles, stroke style, sketch style, solid style, bold style, dashed style.

1 Introduction

People use the free hand drawing to express concepts and to represent ideas in an immediate and intuitive simple way. Therefore, the use of sketch-based interfaces to interact with different human centered systems provides a natural convenient way to carry out concepts and commands. The user interaction is influenced by the individual drawing style. Sometimes, tools used to interact with the different devices (such as: smart phone, PDA, palmtop, tablet-PC, and so on) and/or specific contexts can influence the individual ink style too. Different users can draw the same object/concept using different styles, this can happen without specific reasons because a sketch action is intrinsically an informal and messy human expression. Therefore, the over-tracing and/or cross-tracing phenomena can frequently occur. Figure 1 shows the same object/concept (circle) drawn by different sketches obtained using the three main drawing styles (solid, bold, and dashed). These different drawing styles make the interpretation of the sketch very hard. The purpose of this paper is to propose and discuss an approach to formalize the recognition of the different main styles performed by the user during her/his sketch activity. More specifically, an approach to formalize the recognition of bold, solid and dashed style is presented.

There is an extensive body of related work on sketch style recognition in human centered systems.

B. Apolloni et al. (Eds.): KES 2007/ WIRN 2007, Part II, LNAI 4693, pp. 369–376, 2007.

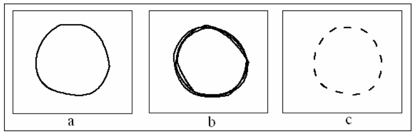


Fig. 1. A circle drawn in: (a) solid style, (b) bold style, (c) dashed style

In [1][2][3] authors suggest a fast simple approach to recognize sketches drawn with a stylus on a digitizing tablet. The approach enables the system to identify shapes (such as: triangles, lines, rectangle, and so on) of different size and with different orientation in the space, drawn with continuous, dashed-tracing or over-tracing strokes. This approach is based on the computation and ratio of particular geometrical features of shapes in the sketch (such as: convex hull, the largest triangle and the largest quadrilateral inscribed in the convex hull, the smallest enclosing rectangle of the convex hull). The over-tracing action, considered as a user imprecision, is discussed in [7]. The system generates geometric approximations for single stroke shapes that are over-traced. According to this approach the system has to match input strokes with geometric primitives producing fits for several shapes (such as: lines, arcs and circles) by computing model's parameters that minimize the least squares fitting error. Also in [9] the problems that involve the interpretation of over-tracing freehand sketch are faced. In this approach over-tracing strokes are interpreted according to the following steps: 1) the strokes are first classified into lines and curves by a linearity test, 2) after this a strokes grouping process that handles lines and curves separately is performed. Afterwards, the grouped strokes are fitted with 2D geometry and further tidied-up with endpoint clustering and parallelism correction. Finally, the in-context interpretation is applied to detect incorrect stroke interpretation based on geometry constraints and to suggest the most probable correction based on the overall sketch context. In [5] the authors present an approach that, unlike the previous two, deals with both the over-tracing and hatching sketch on the 3D sketch. With this approach the strokes are divided into core strokes (strokes that touch the characteristic curves of the object), and hatching strokes (strokes that are mapped to the faces of the object). This approach allows the interpretation of the user's sketch that presents over-tracing strokes, by grouping strokes into bundles. Another approach to recognize the free hand sketches drawn using different styles (with over-traced, dashed or continuous strokes) is given in [8]. The key advance of this approach, is an integrated sketch parsing and recognition model designed to enable natural and pen-based computer interaction. With this approach, the stream of pen strokes is firstly examined to identify delimiter patterns called "markers". These then anchor a spatial analysis, which groups the remaining strokes into distinct clusters, each representing a single visual object. Finally, a shape recognizer is used to find the best interpretations of the clusters. Indeed there are several approaches that face the over-tracing or (less frequently) cross-hatching phenomena. But there are not many works that face the problems from the stroke style point of view. This paper proposes an approach to formalize the recognition of the different drawing styles. The approach does not need to know the features of the geometric shapes in the sketch. It works only on each single stroke. The paper is organized as follows. Section 2 describes both the approach and its formalization to recognize the solid and bold ink styles, Section 3 describes the approach to recognize the dashed ink style. Finally, Section 4 concludes the paper.

2 Recognition and Formalization of Solid and Bold Ink Styles

In this paper a stroke is considered as a drawing action defined by the sequence: pen down, pen movement, pen up. As shown in Figure 2-a, the stroke is spatially characterised by the coordinates (x,y) and temporally characterised by the temporal interval (Δt) used by the user to perform the sketch.

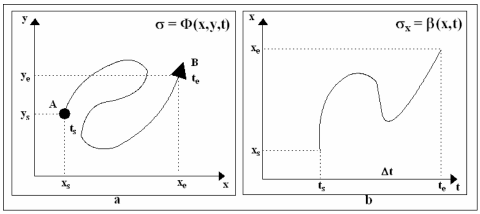


Fig. 2. (a) The stroke σ, (b) The projection, on x, of the stroke σ

In Figure 2-a, the label A shows the start point of the stroke characterized by the start coordinates (x_s,y_s) at the time of pen down t_s. While the label B shows the end point of the stroke characterized by the end coordinates (x_e,y_e) at the time of pen up t_e. For this reason the drawing of a stroke σ can be described as function Φ of both spatial coordinates (x,y) and temporal coordinate (t):

$$\sigma = \Phi(x,y,t) \ . \qquad (1)$$

The first step in detecting stroke style consists of the analysis of the spatial and temporal sequence of the drawn stroke. The aim of analysis is to detect the possible direction change conveyed by the user during stroke drawing. As shown in Figure 2, this analysis can be performed by studying the behavior of a single space component, of the stroke, in function of the time. Considering only one of the two spatial coordinates of each pixel (for instance x) and the time (t) in which the pixels have been drawn, it is possible to obtain a reliable transposition (respect the x axes) of the stroke drawn by the user, that is:

$$\forall t \in \Delta t \ \exists x \in \mathbb{N} : \ \sigma = \Phi(x,y,t) \Rightarrow \sigma_x = \beta(x,t); \text{ where } \Delta t = t_e - t_s \ . \qquad (2)$$

In this way, as shown in Figure 2-b, a function (σ_x) depending on both the chosen axis (x) and the time (t) can be considered. This function represents the projection on x axes of the stroke (σ). The Δt represents the drawing stroke time interval.

The projection function (σ_x) allows highlighting the direction change performed by the user during stroke drawing (σ). In fact, as shown in Figure 3, these changes (on stroke σ) are conveyed in proximity to the relative and/or absolute maximum and minimum points belonging to the projection function (σ_x).

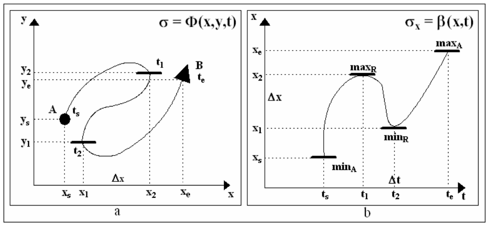


Fig. 3. (a) Stroke with two direction changes, (b) Relative and absolute maximum and minimum points

In this context, the direction changes of the stroke (σ), is defined as the transition of x values from non-increasing monotone to non-decreasing monotone (and vice versa). In Figure 3-b the projection function (σ_x) clearly shows four maximum and minimum points, given below:

$$\begin{aligned}
&\text{absolute minimum point } \min_A \text{ of coordinates } (x_s, t_s) \text{ since}: \forall t \in \Delta t, \exists x \in \Delta x \Rightarrow \beta(x,t) \geq \beta(x_s, t_s)\,.\\
&\text{relative minimum point } \min_R \text{ of coordinates } (x_1, t_2) \text{ since}: \exists CR_t : \forall t \in CR_t \cap \Delta t, \exists x \in \Delta x \Rightarrow \beta(x,t) \geq \beta(x_1, t_2)\,.\\
&\text{relative maximum point } \max_R \text{ of coordinates } (x_2, t_1) \text{ since}: \exists CR_t : \forall t \in CR_t \cap \Delta t, \exists x \in \Delta x \Rightarrow \beta(x,t) \leq \beta(x_2, t_1)\,.\\
&\text{absolute maximum point } \max_A \text{ of coordinates } (x_e, t_e) \text{ since}: \forall t \in \Delta t, \exists x \in \Delta x \Rightarrow \beta(x,t) \leq \beta(x_e, t_e)\,.
\end{aligned} \tag{3}$$

Where CR_t represents a circular round of t, and Δx represent the variability range of the x values during stroke drawing. In particular the points $\max_R$ and $\min_R$ identify the direction changes (respect x axis) of the stroke (σ), while the other two points represent the start ($\min_A$) and the end ($\max_A$) points of the stroke (σ). These four points univocally fix the related coordinates of the stroke (σ). As shown in Figure 3-b, the points $\min_A$, $\max_R$, $\min_R$ and $\max_A$ are respectively related to the points of coordinates (x_s, y_s, t_s), (x_2, y_2, t_1), (x_1, y_1, t_2), and (x_e, y_e, t_e) of the Figure 3-a. The direction changes on stroke (σ) are thus recognized. In Figure 3-a the stroke (σ) shows two direction changes according to the points (x_2, y_2, t_1) and (x_1, y_1, t_2). These changes identify, on the stroke (σ), three different sub-strokes, more exactly:

$$\begin{aligned}
&\text{first sub-stroke fixed by the function}: \sigma = \Phi(x,y,t),\ t \in [t_s, t_1] \subset \Delta t\,.\\
&\text{second sub-stroke fixed by the function}: \sigma = \Phi(x,y,t),\ t \in [t_1, t_2] \subset \Delta t\,.\\
&\text{third sub-stroke fixed by the function}: \sigma = \Phi(x,y,t),\ t \in [t_2, t_e] \subset \Delta t\,.
\end{aligned} \tag{4}$$

The stroke (σ) is subdivided in several strokes, where every identified sub-stroke has a unique direction. The identification of the single sub-stroke is independent from the speed with which the user has drawn it. That is, if the user draws the stroke in Figure 3-a two times with a different speed, the approach will identify the same sub-strokes, as shown in Figure 4.

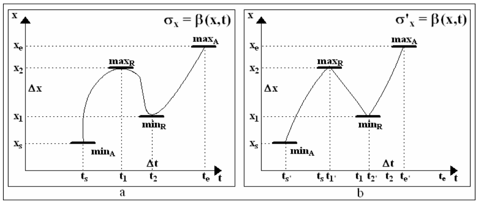


Fig. 4. (a) Transposition of the stroke σ, (b) Transposition of the stroke σ drawn with a different temporal sequences

The user has drawn the stroke which transposition is shown in Figure 4-b faster and keeping a more constant speed than the stroke which transposition is shown in Figure 4-a. Despite of this, the approach has detected (spatially on x axis) the same coordinates for the four maximum and minimum points. Therefore the approach will consider the same three sub-strokes founded for the stroke in Figure 3-a. During sketch activity a stroke can cross itself one or more time, to accomplish the proposed approach each over-traced pixel time has to be considered. When all the sub-strokes have been detected the second step of the approach can be performed.

In the second step, for every established sub-stroke an enclosing rectangle has to be considered, as shown in Figure 5-d, 5-e and 5-f. To perform this result, a reliable transposition, this time respect the y axes, of the stroke has been considered:

$$\forall t \in \Delta t \; \exists y \in \mathbb{N} : \sigma = \Phi(x,y,t) \Rightarrow \sigma_y = \alpha(y,t) \; . \tag{5}$$

Through the projection function (σ_y), as shown in Figure 5-a, 5-b and 5-c, for each sub-stroke can be identified two points: the absolute maximum ((y_1,t_b), (y_3,t_d), (y_5,t_f)) and the absolute minimum ((y_2,t_a), (y_4,t_c), (y_6,t_e)) respect the y axes.

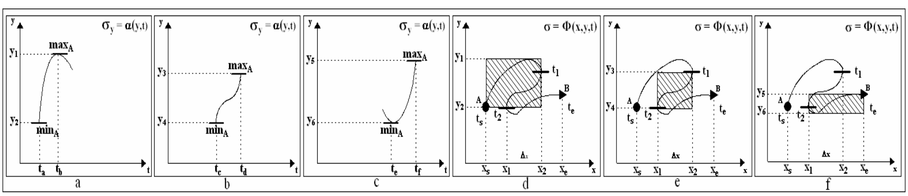


Fig. 5. Absolute maximum and minimum points respect y axes on: (a) first, (b) second, (c) and third sub-stroke. Enclosing rectangles on: (d) first, (e) second, (f) and third sub-stroke.

As shown in Figure 5-d, 5-e and 5-f, for each stroke (first, second and third) the absolute maximum and minimum points respect both x and y axis can be used to identify the connected enclosing rectangles as follow:

$$\begin{aligned} ER_1 &= \int_{y_2}^{y_1}\int_{x_s}^{x_2} \sigma \, d_y d_x = \int_{y_2}^{y_1}\int_{x_s}^{x_2} \Phi(x,y,t) \, d_y d_x \text{ where } t \in [t_s, t_1] \subset \Delta t \, . \\ ER_2 &= \int_{y_4}^{y_3}\int_{x_1}^{x_2} \sigma \, d_y d_x = \int_{y_4}^{y_3}\int_{x_1}^{x_2} \Phi(x,y,t) \, d_y d_x \text{ where } t \in [t_1, t_2] \subset \Delta t \, . \\ ER_3 &= \int_{y_6}^{y_5}\int_{x_1}^{x_e} \sigma \, d_y d_x = \int_{y_6}^{y_5}\int_{x_1}^{x_e} \Phi(x,y,t) \, d_y d_x \text{ where } t \in [t_2, t_e] \subset \Delta t \, . \end{aligned} \tag{6}$$

The last step of the approach to discriminate the stroke style (bold or solid) is to analyze the relationship among the associated areas to the enclosing rectangles (Ar_1, Ar_2 and Ar_3 as shown in Figure 6-a and 6-c). A stroke drawn in a solid style usually has all its associated areas to the enclosing rectangles weakly overlapped, as shown in Figure 6-a and 6-b, while, a stroke drawn in a bold style usually has all its enclosing rectangles strongly overlapped, as shown in Figure 6-c and 6-d, formally:

$$A_{rel} = A_1 \cup A_2 = (A_{tot}) - ((Ar_1 \cap Ar_2) \cup (Ar_1 \cap Ar_3) \cup (Ar_2 \cap Ar_3) \cup (Ar_1 \cap Ar_2 \cap Ar_3)) . \quad (7)$$

Where $A_{tot} = (Ar_1 \cup Ar_2 \cup Ar_3)$ and A_{rel} represents the relationship among associated areas. As it is possible to observe, in Figure 6-b, there are two wide areas (A_1 and A_2) that are covered only by two different associated areas to the enclosing rectangles. Also in Figure 6-d, it is possible to observe two areas (A_1 and A_2) that are covered only by two different associated areas to the enclosing rectangles. The area A_{rel} related to the bold stroke (Figure 6-d) is smaller than one related to the solid stroke (Figure 6-b).

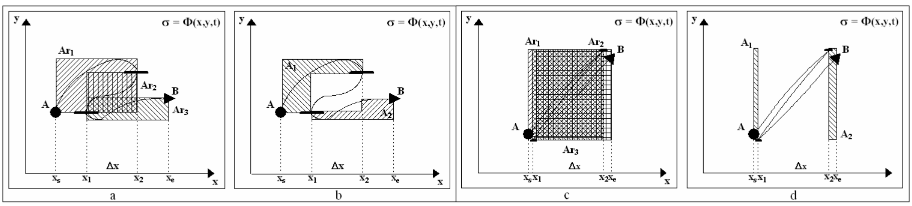
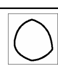
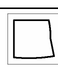
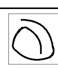
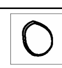
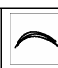
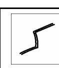


Fig. 6. Enclosing rectangle on: (a)(b) solid stroke, (c)(d) bold stroke

Our training data set were composed by 200 shapes (100 drawn in solid style and 100 drawn in bold style). Each shape has been drawn by a stroke. The experimental results have shown that in a solid stroke the area A_{rel} represents usually a large portion (among the 46% to 100%) of the related area A_{tot}. While, in a bold stroke the area A_{rel} represents usually a small portion (among the 0% to 37%) of the related area A_{tot}. Obviously, it is always possible to draw a stroke where the areas (A_{rel} and A_{tot}) have an ambiguous relationship (among 38% to 45%), it depends on the personal concept of bold and solid style which fixes the ratio of the areas. In table 1 are shown some meaningful examples of shapes style detection.

Table 1. Some visual experimental results of the proposed approach

	SOLID					BOLD				
Single Stroke					..					..
N° Rectangles	5	4	4	3	..	4	8	3	6	..
Ratio(A_{rel},A_{tot})	81%	93%	89%	78%	..	16%	14%	7%	22%	..

The computational complexity required by proposed approach for all basic operations (such as: direction change detection, building rectangles, and so on) can be considered constant. For this reason the whole complexity of the related algorithm can be considered linear in the size of the pixels that make up the stroke σ.

3 Recognition of Dashed Ink Style

The dashed style is composed from more solid and/or bold strokes. As shown in Figure 7-a and 7-b for each one of the five strokes (three solids, two bolds) an enclosing rectangle is considered. Afterwards, the barycentre of each enclosing rectangle is determined and, as shown in Figure 7-c and 7-d, a neighbourhood relationship has to be identified among the rectangles' barycentres. The neighbourhood relationship is based on simple distance measure (fixed threshold, thr) among rectangles' barycentres.

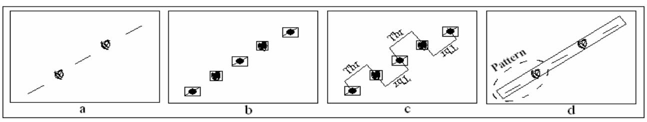


Fig. 7. (a) Five solid/bold strokes, (b) enclosing rectangle and barycentre, (c) threshold, (d) a dashed stroke

Obviously, when two rectangles' barycentres are too far (respect the thr) more than one dashed stroke is considered. Introducing similarity measures (such as the entropy, that measures the disorder degree of a statistical distribution) into the enclosing rectangle could be possible to recognize the specific pattern that made up the dashed stroke. In this case the simple pattern is represented by a single solid stroke and a single bold stroke (as shown in Figure 7-d) but a generic dashed stroke can have a more complex pattern. Finally, as added value, taking into account the temporal information on the strokes that composes a pattern (that is the temporal appearance of a stroke respects the others) it is also possible to identify the verse of a pattern. Combining the shapes into the previous data set has been possible to test the dashed ink style recognition. Obviously, the effectiveness of this approach step depends on the fixed thr, thus if the strokes that compose the pattern are "homogenous" (that is: the enclosing rectangles that compose each stroke of a pattern have a comparable size) than it will always possible to identify the connected pattern.

4 Conclusions

The spread of sketch based interfaces to support the different human centered systems is mainly due to the fact that they make the interaction in different contexts and on different devices intuitive and spontaneous. The ink style through which the user performs a sketch is a critical aspect, therefore recognizing and interpreting steps are hard tasks. The selection of a specific ink style is driven by a personal ink style. Sometimes, it also depends from the interaction tools and/or the specific contexts.

This paper proposes both a strategy and its formalization to detect the stroke ink style (solid, bold and dashed) for sketch based interfaces on human centered systems. The shown approach takes into account only the single stroke and it does not need to know the shapes features contained in the sketch. In this way it is possible to recognize also the different ink styles that can compose the different sides of the same object/shape. The proposed approach discussed starting from the x axis can be extended to the other spatial coordinates.

References

1. Fonseca, M.J., Jorge, J.A.: Experimental Evaluation of an on-line Scribble. Pattern Recognition Letters. 22(12), 1311–1319 (2001)
2. Fonseca, M.J., Jorge, J.A.: Using Fuzzy Logic to Recognize Geometric Shapes Interactively. In: Fuzz IEEE, the 9th IEEE International Conference on Fuzzy System (FUZZIEEE), San Antonio, TX USA, vol. 1, pp. 291–296 (2000)
3. Fonseca, M.J., Jorge, J.A.: A simple approach to recognize geometric shapes interactively. In: Chhabra, A.K., Dori, D. (eds.) GREC 1999. LNCS, vol. 1941, pp. 266–276. Springer, Heidelberg (2000)
4. Fonseca, M.J.: Sketch-Based Retrieval in Large Sets of Drawings. Ph.D. Thesis Department of Information Systems and Computer Engineering. Technical University of Lisbon Superior Technical Institute (July 2004)
5. Mitani, J., Suzuki, H., Rimura, F.: 3D sketch: sketch-based model reconstruction and rendering, pp. 85–98. Kluwer Academic Publishers, Dordrecht (2002)
6. Pratt, W.K.: Digital Image Processing, 3rd edn. John Wiley & Sons, Chichester (2001)
7. Sezgin, T.M., Davis, R.: Handling Overtraced Strokes in Hand-Drawn Sketches. In: Proceedings of the AAAI Spring Symposium Series: Making Pen-Based Interaction Intelligent and Natural, Washington, D.C. (October 21-24, 2004)
8. Kara, L.B.: Automatic parsing and recognition of hand-drawn sketches or pen-based computer interfaces. Ph.D. Thesis Mechanical Engineering Department Carnegie Mellon University (September 2005)
9. Ku, D.C., Qin, S.F., Wright, D.K.: Interpretation of Overtracing Freehand Sketching for Geometric Shapes. In: The 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG ' 2006) (January 30 - February 3, 2006)