二次元スケッチのためのストロークの自動対応付けについての一検討

Ngo Thi Tu Trung　北岡伸也　　北村喜文　　岸野文郎
大阪大学　大学院情報科学研究科

任意の二つの手描きスケッチが与えられた際に，それぞれの特徴に基づいて，スケッチ内のストロークを自動的に対応付ける手法を提案する．提案手法は二つのステップからなる．最初のステップでは，共分散行列を利用し，ストロークの重心や方向などの特徴を抽出する．次のステップでは，反復計算により更新することで，二つのスケッチ間のストロークを対応付ける．提案手法は，モーフィングのための対応付けや，複数のスケッチへの同一アニメーションの適用などに応用できる．

Automatic Stroke Correspondence Detection for 2D Drawings

Ngo Thi Tu Trung　Shinya Kitaoka　　Yoshifumi Kitamura　Fumio Kishino
Department of Multimedia Engineering
Graduate School of Information Science and Technology

We propose a method for detecting automatically stroke correspondences between two drawings. The method composed of two stages: stroke features extraction and stroke correspondence detection. We also conducted an experiment to evaluate the efficiency of our method. This method can be applied to animation generation and morphing field.

1. Introduction

In recent years, many researchers have taken sketch in consideration to generate new sketches by morphing from one input sketch to another input one ([Kort] and [Baxter]) or to recognize Chinese character. In order to obtain good results in sketch morphing and pattern recognition, the right correspondence relation between strokes which make up the sketches is required. However, most of researches on stroke correspondence detection base on an existed database with limited examples or if not, the sketches which users can input are limited (Chinese character recognition). On the other hand, W. Baxter el al. [Baxter] let users input freehand drawings but the method depends strongly on strokes' input order and strokes' connection relation. Therefore, users need to draw with attention and make sure that strokes' connectivity relation between the drawings is correct. This limits users' freedom and contradicts with the intrinsic ambiguous and messy nature of sketch.

We propose a non example-based method to automatically detect stroke correspondences of two drawings without considering strokes' input order and strokes' connection relation. This method allows users to draw more freely comparing with W. Baxter's method [Baxter]. This method composed of two stages: drawings' normalization and stroke correspondences detection. We conducted experiments to test our method's accuracy and discussed the results. The rest of this paper is organized as follows. Section 2 introduces related works. We explain our method's technique in section 3. We evaluate our method's efficiency in section 4 using the experiments' results. Finally, section 5 concludes this paper.

2. Related Work

Related work can be found in two different research fields: stroke's features extraction and stroke correspondence extraction.

Due to the messy and ambiguous nature of sketches, understanding a stroke's features remains a challenging task. B. Pascal et al. [Pascal] analyze each stroke using covariance matrix to find two principal elongation directions of the stroke. T. M. Sezgin et al. [Sezgin] and Hammond et al. [Hammond] detect vertices in a stroke. A stroke is fragmented based on templates in H. Hse's research [Hse]. M. Oltmans [Oltmans] provides a method to understand stroke's overtracing segmentation. Igarashi et al. [Igarashi] extract relations between previously inputted strokes and newly inputted one. However, their system only deals with straight lines. There are also efforts to extract stroke's style such as the vibration, zigzag or jag in the stroke. T. Morimoto et al. [Morimoto] propose a method to extract the vibration in a stroke. Similarly, Freeman [Freeman] and A. Hertzmann [Hertzmann] try to extract one artist's style based on that artist's stroke training set.

In order to generate inbetweenings and animation from drawings, correspondence relation is detected between stroke chains [Kort], between strokes of multiple drawing [Baxter]. Correspondence can be manually specified [Reeves] or found using identical stroke order [Burtnyk]. D. Kurlander et al. [Kurlander] use curve matching for graphical search and replace.

In Chinese character recognition, stroke correspondence is obtained in C. Liu [Liu] using the structural model of character. However, the inter-stroke relations were assigned manually. J. Shin [Shin] employs inter-stroke information, shape and position of each stroke to search stroke correspondence.

3. Automatic stroke correspondence

3.1 Objective drawing's normalization

In almost cases, drawings of same object drawn by various users have different properties in size of whole drawing and each stroke. Therefore, before applying the main correspondence finding algorithm, the scales of strokes and whole drawings need to be normalized. This preprocessing step consists of two small steps. Let N be the number of stroke, each stroke of drawings K_j (j=1,2) be $S_{i,j}$ (i=1,2,…,N). In the first step, each stroke $S_{i,j}$ is transformed to a point $O_{i,j}$ which represents that stroke's center considering the distribution of all points in the stroke. The Principal Component Analysis (PCA) [Pearson] is applied to compute two principal components of $S_{i,j}$ and the bounding box is identified using these components (Fig. 3.1).

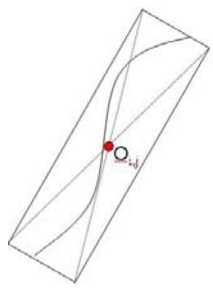

Fig.3.1 Bounding box and center

In the second step, drawings are scaled to fit a unit square. The center point G_j of each drawing K_j is identified based on distribution of $O_{i,j}$. Let O be the origin in the drawing coordinate (X,Y) and $T = \{G_j \mid j=1,2\} \rightarrow O$ be a translation function that move $\{ G_j \mid j=1,2\}$ to O. Then, all points in the set $\{O_{i,j} \mid$

$i=1,2,\ldots,N,\ j=1,2\} \cup \{ G_j \mid j=1,2\}$ are translated using translation formula T. Consider the set of points $\{O_{i,j}\ \ i=1,2,\ldots,N\}$ (j=1,2) and their coordinate (X_{Oij}, Y_{Oij}), $max_{j,X} = \max\{|X_{Oij}| \mid i=1,2,\ldots,N\}$ and $max_{j,Y} = \max\{|Y_{Oij}| \mid i=1,2,\ldots,N\}$ are defined. Let define $F_X(X) = X / max_{j,X}$, $F_Y(Y) = Y / max_{j,Y}$ be the scaling functions for X and Y. By applying the scaling functions to all points $\{O_{i,j} \mid i=1,2,\ldots,N,\ j=1,2\}$, two new set of points $PK_j = \{Q_{i,j} \mid i=1,2,\ldots,N\}$ (j=1,2) where coordinate of each point $Q_{i,j}$ is $X_{Qi,j} = F_X(X_{Oij})$, $X_{Qi,j} = F_Y(Y_{Oij})$ are obtained. Each set of points PK_j represents for drawing K_j.

3.2 Stroke correspondence algorithm

In this research, we propose a method that estimating stroke correspondence by taking into account the similarity of point features and applying the stable marriage algorithm [Gale]. To explain the detail of algorithm let consider some definitions of point features in drawing coordinate and the similarities measure based on those features.

Definition 1: Distance feature vector

Let consider a set of points $PK = \{Q_i \mid i=1,2,\ldots N\}$. Given a set $U \subset PK$, the distance feature vector based on U of a point $Q_k \in PK - U$ ($k \in \{1,2,\ldots,N-|U|\}$) is defined as $DiF(Q_k|U) = (d(Q_k,Q_j) \mid Q_j \in U)$, where $d(Q_k,Q_j)$ is Euclid distance between Q_k and Q_i.

Definition 2: Angle feature vector

Similar to Definition 1, given a set $U \subset PK$, the angle feature vector of a point $Q_k \in PK - U$ ($k \in \{1,2,\ldots,N-|U|\}$) is defined as $AgF(Q_k|U) = ((OQ_k,OQ_i) \mid Q_j \in U)$, where O is origin of drawing coordinate and (OQ_k,OQ_i) is angle between OQ_k and OQ_i.

Definition 3: Distance feature similarity

Let consider two points Q_k and Q_k‘, and their distance feature vector $DiF(Q_k|U)$ and $DiF(Q_k‘|U’)$. U and U’ are different and have same cardinality (|U| = |U’|). The distance feature similarity of Q_k and Q_k‘ is defined as SDiF $(Q_k,Q_k‘|U,U’) = d(DiF(Q_k|U), DiF(Q_k‘|U’))$ where, d is Euclid distance between two points in coordinate of distance feature vector DiF.

Definition 4: Angle feature similarity

Using same assumption to Definition 3, the distance feature similarity of different points Q_k and Q_k‘ is defined as SAgF $(Q_k,Q_k‘|U,U’) = d(AgF(Q_k|U), AgF(Q_k‘|U’))$ where, d is Euclid distance between two points in coordinate of angle feature vector AgF.

Definition 5: Mix feature similarity

For two point Q_k and Q_k‘, given distance feature similarity SDiF $(Q_k,Q_k‘|U,U’)$ and SAgF $(Q_k,Q_k‘|U,U’)$, the mix feature similarity is defined by MF = d(SDiF $(Q_k,Q_k‘|U,U’)$, SAgF $(Q_k,Q_k‘|U,U’)$), where d is Euclid distance.

The stroke correspondence estimating algorithm is simply described as follows.

Procedure:

Strokes Correspondence Estimate (SCE)

Input

-Two set of points PK_1 and PK_2:

Set $PK_1 = \{Q_{i,1}| i=1,2,\ldots N\}$

Set $PK_2 = \{Q_{i,2}| i=1,2,\ldots N\}$

-Origin of coordinate: O.

Set $U_1 = \{O\}$, $U_2 = \{O\}$

Step 1

-For i=1,2,…N, j = 1,2,..N

Compute SDiF $(Q_{i,1} , Q_{j,2} |U_1,U_2)$.

End

- For i=1,2,…N

-Create $L_{i,1} = \{ Q_i^k{}_{,2} \mid k=1,2,\ldots N \}$, where

$SDiF(Q_{i,1},Q_i^k{}_{,2}|U_1,U_2) < SDiF(Q_{i,1},Q_i^{k+1}{}_{,2}|U_1,U_2)$

-Create $L_{i,2} = \{ Q_i^k{}_{,1} \mid k=1,2,\ldots N \}$, where

$SDiF(Q_{i,2},Q_i^k{}_{,1}|U_1,U_2) < SDiF(Q_{i,2},Q_i^{k+1}{}_{,1}|U_1, U_2)$

End

-Apply StableMarriage($L_{i,1}$, $L_{i,2}$|i=1,2,..N) to obtain a set of corresponding pairs.

$R_{dis} = \{(Q_i^p{}_{,1} , Q_j^p{}_{,2}) \mid p=1,2,..N\} \ (i^p,j^p \in \{1,2,\ldots N\})$

Step 2

Similar to step 1, by using angle feature similarity, SAgF ($Q_{i,1}$, $Q_{j,2}$ |U_1,U_2) (i,j=1,2,…N) the other set of corresponding pairs is obtained..

$R_{ang} = \{(Q_i^a{}_{,1} , Q_j^a{}_{,2}) \mid a=1,2,..N\} \ (i^a,j^a \in \{1,2,\ldots N\})$

Step 3

Similar to step 1, by using angle feature similarity, MF ($Q_{i,1}$, $Q_{j,2}$ |U_1,U_2) (i,j=1,2,…N) the other set of corresponding pairs is obtained..

$R_{mix} = \{(Q_i^m{}_{,1} ,Q_j^m{}_{,2}) \mid m=1,2,..N\} \ (i^m,j^m \in \{1,2,\ldots N\})$

Step 4

Find set of pairs $R = R_{dis} \cap R_{ang} \cap R_{mix}$

If R = 0 then

Set $R = R_{mix}$

Set $R_{final} = R_{final} \cup R$

Stop the algorithm FCE.

else

Set $R_{final} = R_{final} \cup R$

Set $U_1 = \{Q_i^{fin}{}_{,1} \mid Q_i^{fin}{}_{,1} \in R_{final} \}$

Set $U_2 = \{Q_j^{fin}{}_{,2} \mid Q_j^{fin}{}_{,2} \in R_{final} \}$

Set $PK_1 = PK_1 - U_1$

Set $PK_2 = PK_2 - U_2$

Repeat

Procedure SCE(PK_1 ,PK_2 ,U_1 ,U_2)

End

Our proposed algorithm SCE has 4 steps. Step 1, 2, 3 conducts SCE using distance feature similarity, angle feature similarity, mix feature similarity, respectively. In step 4, SCE is repeated using new set of points U_1, U_2 in corresponding pair set R_{final}.

If we use only distance feature similarity or angle distance similarity, the accuracy becomes low due to cases in which distance feature similarities are close but angle feature similarities are much different and vice versa. Using mix feature similarity helps remove wrongly corresponded couples.

4. Evaluation

4.1.Experiment Design

We conducted experiments using a test program on a Pentium M 1.8GHz 512MB of Ram laptop. We created database of drawings beforehand and called each of 20 drawings randomly in each experiment. The drawings are different in stroke number. We have 6 subjects draw drawings similar to the ones shown by the program. The word "similar" here means that drawings which subjects draw have same stroke number and structure with the ones they see.

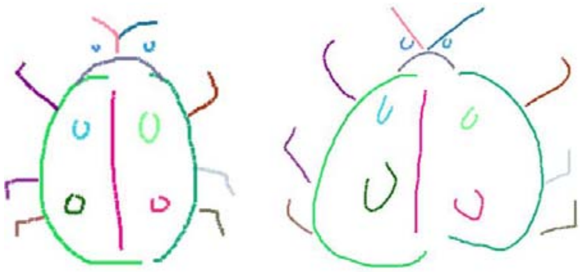

Fig.4.1 An example of experiment's results. On the left is database's drawing, on the right is user-drawn drawing.

Fig. 4.1 shows an example with left drawing is database's one and the right drawing is subject-drawn one. After drawing, correspondence is automatically detected and strokes with same color are shown as corresponded ones. Each subject will be asked to count wrongly-corresponded stroke number. This is based on user's intuition**.**

Subjects are students with age ranges from 22 to 28 and more than 5 years of computer experience. Most of them have less experience with doodle.

4.2 Result and Discussion

Data collected from subjects is calculated to find average right correspondence rate for each stroke number. We plotted a graph to show the relation of stroke number and right correspondence rate (Fig.4.2). From the result, drawing with stroke number from 3 to 6, which is simple has high right correspondence rate. This can be explained due to two reasons. First, the simpler the drawing is, the more easily user can draw. Second, because our algorithm bases largely on position of stroke's

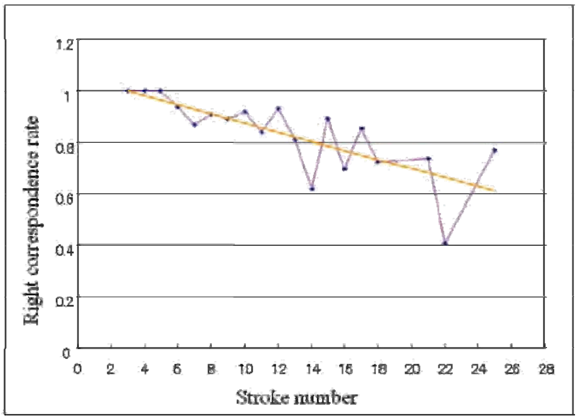

Fig.4.2 Relation of stroke number and right correspondence rate

center in the drawing, the less stroke number is, distances between stroke's centers are more different and stroke correspondence can be correctly extracted. When stroke number increases, right correspondence rate decreases but still oscillates between 0.7 and 0.94. In 14 and 22 cases, rate sudenly decreases. Based on users' comments, we consider small sample number (comparing to other cases) and messy drawings are the reasons. Our algorithm's accuracy falls when the drawing has many strokes which distribute densely like the legs in Fig.4.3.

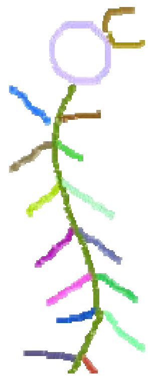

Fig.4.3 Case with low correspondence rate

5. Conclusion

In this paper, we propose a method to automatically detect stroke correspondence in 2D drawings in relatively ignorable computation time. The method allows users to draw relatively freely with accuracy rate ranges from 0.7 ~ 1. In the future, we plan employ stroke's structure information to improve accuracy.

References

[Sezgin] T. M. Sezgin, T. Stahovich, and R. Davis: Sketch Based Interfaces: Early Processing for Sketch Understanding. *Perceptive User Interfaces Workshop* (2001).

[Hertzmann] A. Hertzmann, N. Oliver, B. Curless, and S. M. Seitz: Curve Analogies. *Proc. of the 13th Eurographics workshop on Rendering*, pp. 233--246 (2002).

[Igarashi] T. Igarashi, S. Matsuoka, S. Kawachiya, and H. Tanaka: Interactive Beautification: *A Technique for Rapid Geometric Design. Proc. ACM UIST*, pp. 105--114(1997).
[Kort] A. Kort: Computer Aided Inbetweening. *Proc. of the 2nd international symposium on Non-photorealistic animation and rendering*, pp. 125--132 (2002).
[Baxter] W. V. Baxter and K. Anjyo: Latent Doodle Space. *Computer Graphics Forum (Eurographics 2006)*, Vol. 25, Issue 3, pp. 477--485 (2006).
[Pascal] B. Pascal, B. Simon, T, Jolle, S. Francois, and L. Markosian: Stroke pattern analysis and synthesis. *Computer Graphics Forum (Proc. of Eurographics 2006)*, Vol. 25 (2006).
[Morimoto] T. Morimoto, S. Takagi, and F. Yoshimoto: 心地よい手書き風曲線の自動生成. *情報処理学会関西支部_新・支部大会_*CGセッション, A-15 (2002).
[Freeman] W. T. Freeman, J. B. Tenenbaum, and E. Pasztor: An example-based approach to style translation for line drawings. *Technical Report TR99-11, MERL* (1999).
[Hse] H. Hse, M. Shilman, A. R. Newton: Robust sketched symbol fragmentation using templates. *Proc. of the 9th international conference on Intelligent user interfaces IUI '04*, pp. 156—160 (2004).
[Oltmans] M. Oltmans: Envisioning sketch recognition: a local feature based approach to recognizing informal sketches. *Massachusetts Institute of Technology* (2007).
[Hammond] T. Hammond, B. Eoff , B. Paulson, A. Wolin, K. Dahmen, J. Johnston, and P. Rajan: Free-sketch recognition: putting the chi in sketching. *CHI '08: CHI '08 extended abstracts on Human factors in computing systems*, pp. 3027--3032 (2008).
[Kurlander]D. Kurlander and E. A. Bier: Graphical search and replace. *Proc. of the 15th SIGGRAPH Computer Graphics*, pp. 113--120 (1988).
[Liu] C. Liu, I. Kim, J. H. Kim: Model-based stroke extraction and matching for handwritten Chinese character recognition. *Pattern Recognition*, Vol. 34, pp. 2339—2352 (2001).
[Reeves] W. T. Reeves: Inbetweening for computer animation utilizing moving point constraints. *Proc. of the 8th SIGGRAPH Computer Graphics*, Vol. 15, pp. 263--269 (1981).
[J-P Shin] J-P Shin: Optimal stroke-correspondence search method for on-line character recognition. *Pattern Recognition Letters*, Vol. 23, Issue 5, pp. 601--608 (2002).
[Burtnyk] N. Burtnyk and M. Wein: Computer animation of free form images. *Proc. of the 2nd SIGGRAPH Computer Graphics*, Vol. 9, Issue 1, pp. 78--80 (1975).
[Pearson] K. Pearson: On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine 2* (6), pp. 559--572.(1901).
[Gale] D. Gale and L. S. Shapley: College Admissions and the Stability of Marriage. *American Mathematical Monthly* , 69, 9-14 (1962).