Journal of Southeast University (English Edition) Vol. 21, No. 2, pp. 153 - 158 June 2005 ISSN 1003—7985

Online composite shape recognition based on relevance feedback

Wang Qiang Sun Zhengxing

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

Abstract: This paper describes a novel method of online composite shape recognition in terms of the relevance feedback technology to capture a user's intentions incrementally, and a dynamic user modeling method to adapt to various users' styles. First, the relevance feedback is adapted to refine the recognition results and reduce the ambiguity incrementally based on the establishment of a feature-based vector model of a user's sketches. Secondly, a dynamic user modeling is introduced to model the user's sketching habits based on recording and analyzing historical information incrementally. A model-based matching strategy is also employed in the method to recognize sketches dynamically. Experiments prove that the proposed method is both effective and efficient.

Key words: sketchy-based user interface; online composite shape recognition; dynamic user modeling; relevance feedback

Freehand sketching is a natural and crucial part of design[1]. However, it is difficult to ask a computer to completely understand various sketches or to emulate the real pen-paper interaction pattern according to a user's sense of shape and thought habits.

In fact, the intentions of different users are various, may be inconsistent with similar sketches, and the sketching styles and preferences of the same user may be different at different times. This user diversity partly results in the "ambiguity" in sketching, which brings out many difficulties. These difficulties can be ascribed to the lack of coincidence between the information that one can extract from the visual data of a sketch and the interpretation that the same data has for a user in a given situation. Therefore, benefiting from advances in sketching recognition systems cannot be expected before the problems of ambiguity and user adaptation[2] are well solved.

In this paper, we comprehensively introduce an approach to online composite sketchy shape recognition based on relevance feedback technology, which comes from a textual search system[3]. Besides relevance feedback, we propose that the recognition engine incrementally and continuously collects and analyzes users' subjective judgments of sketch recognition and automatically adjusts the recognition model by relevance feedback. Meanwhile the engine dynamically models a user's profile to deal with user diversity based on historical sketchy recognition information, called the incremental sketch recognition, which intends to fulfill the computability of freeform sketches without requiring extra cognitive load on users. The approach includes several parts: initial shape-recognition, relevance feedback and dynamic user modeling.

1 Framework of Proposed Approach

We develop a framework of composite shape recognition as shown in Fig. 1. First of all, the initial shape-recognition recognizes primitive shapes such as ellipse, arc and straight line, then extracts feature vectors of a sketch in order to construct vector-model for the sketch. Based on vector model, the recognition engine calculates similarity between the sketch and standard shapes, which can result in a candidate objects set. Secondly, the relevance feedback is adopted, which adjusts the vector model incrementally by reforming the feature vector and re-weights the distance function according to the user's judgment and original candidates. Then we redo similarity calculations based on the new vector model, which can result in a new candidate set closer to the user's input intentions. Finally, we apply a dynamic user modeling method, based on an incremental "historical information record and analysis" method, to a historic dataset to model the user's sketching styles and habits on the sketchy shape. A model-based matching strategy is also used.

In concrete terms, the initial shape recognition includes primitive recognition, feature extraction and templates-based matching. The relevance feedback part

Received 2004-07-02.

Foundation items: The National Natural Science Foundation of China (No. 69903006, 60373065), Program for New Century Excellent Talents in University.

Biographies: Wang Qiang (1982—), male, graduate; Sun Zhengxing (corresponding author), male, doctor, professor, szx@ nju. edu. cn.

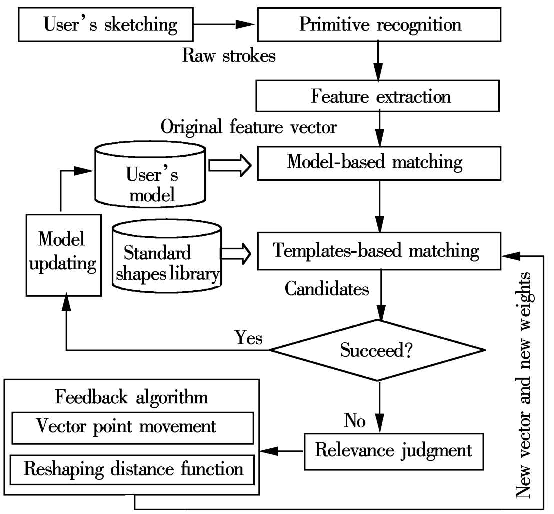

Fig. 1 Framework of proposal recognition approach

includes relevance judgment and a feedback algorithm. Dynamic user modeling partly consists of model updating as well as model-based matching.

2 Initial Shape Recognition

2.1 Primitive recognition

In order to calculate the structural similarity between two composite shapes, we first identify their primitive components: lines, arcs, and ellipses. The primitives of a sketched object are recognized in the primitive recognition stage. Polygons are broken down into lines. Two line/arc segments may be merged as one if they overlap each other or lay side by side very closely. The primitive recognition stage includes stroke pre-process, shape classifying, shape fitting and rectifying, which has been extensively discussed in Ref. [4]. We will not discuss it any further here.

2.2 Feature extraction

For introducing relevance feedback into sketchy shape recognition, a recognition model is designed, which uses an instance of a single feature representation f as a vector point p in a multidimensional space. In our research, features of shape are refined into two types: the edge feature and the spatial relation feature.

The edge feature represents edge type information of a composite shape; we define 16 types of edges according to edges' main direction as shown in Fig. 2 (a). The spatial relation feature between edge (i) and edge(j) is defined as $S(i, j)$. All these relations of a sketchy or standard shape can express the configuration of the shape. We consider eight types of relations between a pair of geometric primitives as shown in Fig. 2(b).

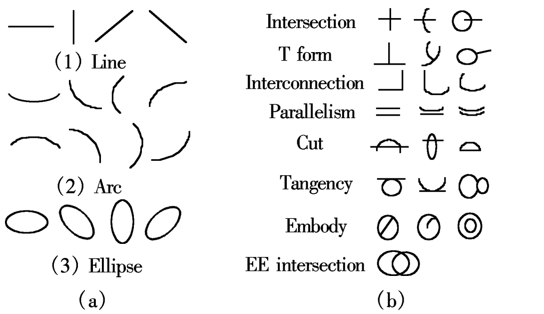

Fig. 2 16 types of edges and samples for 8 types of relation

Therefore, a sketchy or standard shape can be represented by a 16-dimensional vector $\boldsymbol{E}$, where each element $E(i)$ represents the number of type i edge in the shape, and an 8-dimensional vector $\boldsymbol{R}$, where each element $R(i)$ is the number of spatial relation type i the shape contained. For simplification, we concatenate vector $\boldsymbol{E}$ and vector $\boldsymbol{R}$ to a single 24-dimensional feature vector $\boldsymbol{V}$ that finally represents a user's sketch or a standard shape.

Supposing n is the number of edges for an object, the space usage of feature extraction is used to store the feature vector $\boldsymbol{V}$ that can be omitted. Edge feature extraction should traverse all the edges in shape. Meanwhile, relation feature extraction needs C_n^2 comparisons between two edges. Therefore the maximum time complexity is $O(C_n^2 + n)$. The method has high efficiency, because it only considers the edge type and edges relation, which is simpler than whole spatial information of the sketch that is almost an NP problem and complex.

2.3 Templates-based matching

This stage calculates similarity between a user's sketchy shape and standard shapes in library, which results in an objects set, based on the vector-based model.

We employ Euclidean distance for matching, in which the distance between two objects A and B is the weighted sum of all dimensions between them. That is

$$\operatorname{dis}(A, B) = \sqrt{\sum_{i=1}^{24} w_i (V_A(i) - V_B(i))^2} \qquad (1)$$

Before calculating, all the weights w_i have been initialized to the same value 1/24; and all intra-feature attributes have been normalized to make sure that each dimension fall within the same range.

Let n be the total number of edges in a shape, cn be the distance threshold for the feature vector. The similarity between A and B can then be calculated, derived from the Euclidean distances as follows:

$$\mathrm{sim}(A,B) = \begin{cases} 0 & \text{if } \mathrm{dis}(A,B) \geqslant cn \\ 1 - \dfrac{\mathrm{dis}(A,B)}{cn} & \text{else} \end{cases} \tag{2}$$

where c is the constant value. This normalization is simple and effective, which makes $\mathrm{sim}(A, B)$ fall in range $[0, 1]$.

Besides the objects resulting from templates-based matching, the model-based matching stage, which will be discussed in section 4, will generate another set of objects. The recognition engine then combines the results of the two stages to a whole set of candidate object.

3 Relevance Feedback

After the candidate objects are generated, the user looks at individual candidate results and determines whether there is a perfect object that satisfies him completely. If the user specifies an object to be the final result of recognition, it means that the recognition has been successful, otherwise, the user judges whether the result is a good or bad instance of his information need. The feedback algorithm uses the original results and the user's feedback to refine the results incrementally.

3.1 Relevance judgment

For relevance judgment, the user examines the candidate shapes and provides a judgement as to the quality or relevance of the candidates. The user supplies relevance feedback by judging goodness/badness of results. He can provide relevance feedback at varying granularities. Our method supports a binary approach to relevance: a result is either relevant or not. We will denote the candidate objects by a_i, where i indicates the rank of that result, that is, results are ordered based on i: $\langle a_1, a_2, \ldots \rangle$. We denote the relevance feedback for result a_i by f_i. This is a numeric value with the following interpretation:

$$f_i = \begin{cases} 1 & \text{relevant} \\ 0 & \text{no-information} \\ -1 & \text{non-relevant} \end{cases}$$

3.2 Feedback algorithm

The feedback algorithm uses the original candidate results and the user's supplied feedback to capture the user's intentions incrementally, including two primary techniques: vector point movement and reshaping distance function, which is similar to principles of relevance feedback in CBIR[5].

3.2.1 Vector point movement

The objective of the vector point movement approach is to construct a new vector point that is "close" to relevant results, and "far" from non-relevant results. Fig. 3 shows how this approach works. The best-known approach to achieving vector point movement is based on the formula initially developed by Rocchio[3] in the context of textual information retrieval.

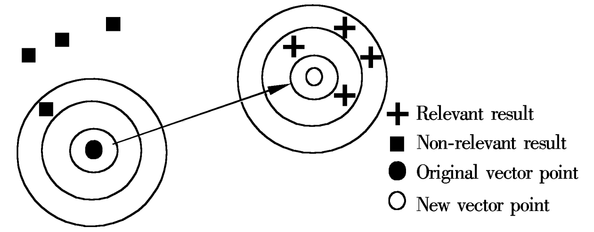

Fig. 3 Vector point movement

Let $S_{\mathrm{rel}} = a_i \mid f_i > 0$ be the set of relevant points user feedback, and $S_{\text{non-rel}} = a_i \mid f_i < 0$ be the set of points the user explicitly marked as non-relevant. The new vector point is an incremental change over the original vector point, which is moved towards the relevant points and away from the non-relevant points:

$$P_{\text{new}} = \alpha P_{\text{old}} + \frac{\beta}{|S_{\text{rel}}|}\sum_{a_l \in S_{\text{rel}}} a_j - \frac{\gamma}{|S_{\text{non-rel}}|}\sum_{a_l \in S_{\text{non-rel}}} a_j \tag{3}$$

The speed at which the old vector point is moved is determined by the parameters α, β, and γ, where $\alpha + \beta + \gamma = 1$. The purpose of retaining part of the original vector point is to avoid "overshooting". The element of the feature vector a_i is $V(i)$. The main advantage of this approach is its simplicity and generally good results. It is intuitive to understand and closely mimics what a human user would do to improve a result, that is, restate the recognition with a different vector.

3.2.2 Reshaping distance functions

While using the vector point movement technique, there are many ways in which similarity calculation can be influenced. Indeed, there is no restriction on the kind of distance function we can use, and its "shape" can be distorted in any arbitrary way that makes sense for that function. One approach to changing the shape of the distance function is to update the weight for each dimension in the distance function. The interpretation of this is to give more importance to certain elements of the feature representation, for example, the intersection relation in a sketch may be more important to the user than the interconnection in the sketch. Fig. 4 shows how a standard Euclidean distance function changes when a weight is given for each dimension.

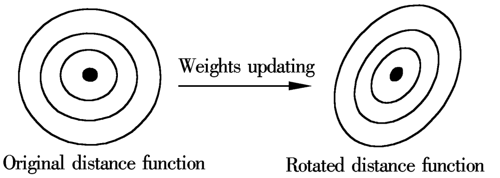

Fig. 4 Distance function reshaping

To derive a new distance function, we change the weights w_i to new values that better capture the user's information need. We suggest choosing weights w_i proportional to the inverse of the standard deviation among the relevant values of dimension i. The intuition behind using standard deviation or variance is that a large variation among the values of good results in a dimension means that dimension poorly captures the user's information need and vice versa. We derive a new distance function shape through the following steps. First we estimate the new weights:

$$w_i^{\text{est}} = \frac{1}{\sigma(a_j(i) \mid a_j \in S_{\text{rel}})} \tag{4}$$

Next, we normalize these weights to ensure they add up to 1 and are compatible with the original weights. The final step is to combine the estimated weights with the original weights: $w_i^{\text{new}} = \alpha w_i + \beta w_i^{\text{est}}$, where parameters α and $\beta(\alpha + \beta = 1)$ control the speed or aggressiveness of the relevance feedback, that is, how much of the original vector weight is preserved for the new iteration.

3.2.3 Recalculation

After vector point movement and reshaping the distance function, we can do a similarity calculation with new feature vector and weights, then we can obtain new candidates' results after recalculation, which are closer to the intentions of user. During iterations of feedback, results are refined incrementally, which can capture the user's intention finally.

4 Dynamic User Modeling

To record users' habits of sketching, a dynamic user modeling is used, where a model updating method is adopted to record and analyze the user's historical information incrementally. We also employ a model-based matching strategy to capture the user's intentions directly without additional feedback.

4.1 Definition of user model

The user model mainly stores the user's subjective intentions for a particular sketch and the optimized weights that reflect the user's drawing styles. To construct user models, the following definitions are given.

Definition 1 One recognition is defined as a 3-tuple $h = (S_o, S_r, t)$, where S_o is represented by the feature vector $\boldsymbol{V}$, S_r is the final result specified by users represented by a shape ID, and t is the draw time of the sketch.

Definition 2 The list of the historical recognition records is expressed as $l = \langle h_1, h_2, \ldots, h_n \rangle$, where h_i is one of the historical records. The list is obtained and updated by model updating incrementally.

Definition 3 A single user model can be denoted as $T = (l, \boldsymbol{w})$, where l contains all the historical records of sketch recognition for the user, and $\boldsymbol{w} = \{w_1, w_2, \ldots, w_{24}\}$, where w_i denotes the corresponding weight for the particular dimensions of $\boldsymbol{V}$. The weights are updating dynamically by model updating.

Definition 4 Multi-user models are composed of many single user models and denoted as $U = (M_1, M_2, \ldots, M_n)$.

4.2 Dynamic model updating

Corresponding to the two parts of the user model, the model updating stage includes two steps as well: records appending and weights updating. Both are done according to the user's judgment. It is necessary to give emphasis to the fact that the updating will happen if and only if the recognition is successful.

4.2.1 Records appending

When candidate shapes are presented to the user, he will designate "relevant", "non-relevant" or "excellent" for each candidate. Once the user specifies the "excellent object", the engine will do the recording. It adds a record $h = (S_o, S_r, t)$ in the user model as in definition 1.

Incidentally, to reduce the data storage in the user model, we dynamically delete records when the lifetime of the record is greater than a time threshold, which can be calculated by comparing the current time and the time value in the record, that is, the user model saves the recent information which can reflect the user's time, habits and intentions.

4.2.2 Weights updating

The initial weights for the similarity calculation saved in the model are typically for the same value. Weights updating can capture the user's sketching habits incrementally and dynamically by analyzing historical recognition information. For weights updating, the difference for dimension i between the original sketched object S_o and final result S_r is

$$\text{diff}_i(S_o, S_r) = \left| \frac{V_{S_o}(i) - V_{S_r}(i)}{V_{S_o}(i)} \right| \quad (5)$$

We suggest updating weights w_i proportional to the inverse of the average of difference. The intuition is that a large average difference in a dimension means that dimension poorly captures the user's intention and should carry a lower weight and vice versa. That is

$$w_i^{\text{update}} = \frac{n}{\sum_{j=1}^{n} \text{Diff}_i(h_{S_o}^j, h_{S_r}^j)} \quad (6)$$

where h_j is the historical record in l defined as above, and n controls the frequency of update.

4.3 Model-based matching

A model-matching scheme is introduced before templates-based matching. When a feature vector has been constructed from feature extraction, we match the drawing sketched to the records in the user's model that may capture the user's intentions directly. Actually, similar sketchy shapes drawn by the same user at different times will be closer in vector space and may have some final result. Therefore, we choose the results in the user model whose feature vectors are closer to that of the current sketch, to be candidate results of recognition.

Which result is chosen to be a candidate depends on the distance between current sketch S_c and record h:

$$\text{dis}(S_c, h) = \delta^{(\text{currenttime} - h_t)} \sqrt{\sum_{i=1}^{24} w_i (V_{S_c}(i) - V_{h_{S_o}}(i))^2} \quad (7)$$

where weight w_i is that saved in the user model, and parameter δ controls the importance of a record's lifetime. It is obvious that the smaller the distance is, the closer the result in h is to the user's current intentions. Therefore, if dis(S_c, h) is smaller than a threshold, we choose the result S_r in h as a candidate result or object for recognition. After matching to all the records in model, it can result in a set of candidate objects probably capturing the user's habit and intents directly.

5 Experiments and Evaluation

We use time t and recognition-rate scale r to evaluate the recognition. t denotes the recognition speed without regard to time consumption for the user's judgments. r is the ratio of the number of successful recognitions to the total number of recognitions. All experiments are done on an Intel P4 PC (with a 2.8 GHz CPU and 512 MB memory) running Microsoft Windows XP. The experiments operate on 350 standard electrical and electronics symbols. We have done two experiments, one tests the effectiveness of relevance feedback, while the other tests the user's model.

For the first experiment, we asked six random persons to draw sketches. Every person drew 10 sketches, and we used our recognition engine to do recognition compared with the SRG method[6]. We computed the recognition time of nine random recognitions. Besides this, the average r after iterations was accounted. We have obtained above experimental results shown in Tab. 1 and Fig. 5.

Tab. 1 Time usage of two methods for nine recognitions ms

Method	Times								
	0	1	2	3	4	5	6	7	8
SRG	32.0	4.7	226.0	15.3	31.2	42.7	6.2	78.2	23.2
Rel-FB	1.3	1.4	1.7	1.4	1.3	1.6	2.1	1.1	1.4

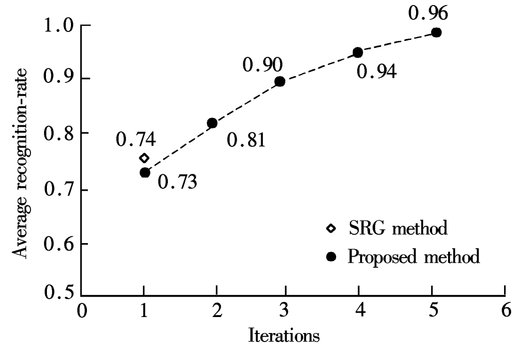

Fig. 5 Recognition-rate r for two methods

For using graph-matching mechanisms resulting in an NP problem, the SRG method has higher time cost, which greatly varies with different recognitions. And the SRG method uses a precise matching, which may lose user's intentions, inducing a lower r. As shown in Tab. 1, the time usage of our method is very small and very steady, which makes it particularly suitable for use online. As shown in Fig. 5, the r of our method before feedback is slightly worse than the SRG method, however, the quality of recognition is obviously improved during four iterations of feedback, especially in iterations 1 and 2.

Secondly, we asked two persons to do sketch recognitions. Each person did sketching of 10 particular different symbols over several days with our dynamic user's modeling recognition method, 20 times sketching each shape. After every 5 times sketching each shape, we calculated the average recognition-rate, we obtained the results for person No. 1 and person No. 2 as shown in Tab. 2 and Tab. 3, respectively. They show that, along with the increase of drawing times for every shape, the average r is growing up gradually. After 20 times drawing, the r before feedback iteration can reach 90%. That is, about 90% of recognitions are successful without the user's feedback; burden of judgment is reduced remarkably. From the tables, we can see that the increase of r for person No. 1 is more irregular than that for person No. 2. It implies that the sketching habits of person No. 1 varied frequently with regard to that of person No. 2 and that the user whose sketch habits is consistent will get more benefits from dynamically user's modeling.

Tab. 2 Experimental results for person No. 1

Draw times	Average r in different iteration				
	0	1	2	3	4
1 to 5	0.76	0.80	0.82	0.88	0.92
6 to 10	0.88	0.93	0.98	1.00	
11 to 15	0.88	0.90	0.90	0.94	0.96
16 to 20	0.90	0.92	0.96	0.98	0.98

Tab. 3 Experimental results for person No. 2

Draw times	Average r in different iteration				
	0	1	2	3	4
1 to 5	0.67	0.78	0.83	0.90	0.95
6 to 10	0.80	0.83	0.85	0.98	0.98
11 to 15	0.90	0.93	0.95	1.00	
16 to 20	0.93	0.97	0.98	1.00	

6 Conclusion

The main challenge in the area of shape recognition comes from the ambiguity and issue of user adaptation in sketch recognition. Relevance feedback as well as user modeling is a valuable tool to reduce ambiguity and the issue of user adaptation. We have exploited an online composite shape recognition method, which abstains from using graph-matching technology that is an NP problem and not adaptable to online recognition, and mainly employs relevance feedback technology and a dynamic user modeling approach, to make an engine shielding the diversity from different thought modes and input habits of different users. Experimental results show the advantages of the proposed method.

References

[1] Fish J, Scrivener S. Amplifying the mind's eye: sketching and visual cognition[J]. *Leonardo*, 1990, **23**(1): 117 − 126.

[2] Sun Z X, Liu W Y, Peng B B, et al. User adaptation for online sketchy shape recognition [A]. In: *Proceedings of GREC Conference*[C]. Barcelona, Spain, 2003. 305 − 315.

[3] Rocchio J J. Relevance feedback information retrieval [A]. In: Salton G, ed. *The Smart Retrieval System: Experiments in Automatic Document Processing* [C]. Englewood Cliffs, NJ: Prentice Hall, 1971. 313 − 323.

[4] Sun J Y, Jin X Y, Pen B B, et al. A method of fast on-line graphics recognition and regularization [J]. *Computer Science*, 2003, **30**(2): 172 − 176.

[5] Aigrain O H, Zhang H, Petkovic D. Content-based representation and retrieval of visual media: a state of the art review [J]. *Multimedia Tools and Applications*, 1996, **3**(1): 179 − 202.

[6] Xu X G, Sun Z X, Liu W Y. Matching spatial relation graphs using a constrained partial permutation strategy [J]. *Journal of Southeast University (English Edition)*, 2003, **19**(3): 236 − 239.

基于相关反馈技术的在线复杂图形识别

王　强　　孙正兴

(南京大学计算机软件新技术国家重点实验室, 南京 210093)

摘要: 提出一种在线复杂图形识别方法, 在该方法中引入相关反馈技术来逐渐捕获用户绘图意图并使用用户模型来适应不同用户的绘图习惯. 首先, 在根据图形向量化特征的相似度计算给出候选识别结果的基础上, 利用相关反馈技术不断降低模糊性而提高识别效果. 其次, 通过记录和分析"历史信息"动态的为不同用户建立用户模型, 从而适应不同的用户习惯和绘图意图. 同时, 引入了一个基于模型的动态匹配策略. 实验证明所介绍的识别方法在试验中取得了很好的效果.

关键词: 基于草图的用户接口; 在线复杂图形识别; 动态用户模型; 相关反馈

中图分类号: TP391