

Sketch Recognition Based on Topological Spatial Relationship

Binbin Peng, Yin Liu, Liu Wenyin, and Guanglin Huang

Dept. of Computer Science, City University of Hong Kong, Hong Kong SAR, PR China
{cspengbb, liuyin, csluwy}@cityu.edu.hk
hwanggl@cs.cityu.edu.hk

Abstract. In recognition of composite graphic objects, topological spatial relationships of their components play an important role. Although most researchers focus on binary topological relationships, they cannot carry all information of the internal structure of the compound objects. Therefore, we introduce the ternary relationship, which is a complement to the binary relationship, to describe composite graphic objects. Moreover, we provide a constrained partial permutation algorithm based on both the binary and ternary topological spatial relationships to recognize the sketchy objects input by users in an online manner. Experimental results show that this approach is both efficient and effective for online composite graphics recognition in our sketch-based graphics input system – SmartSketchpad.

1 Introduction

Sketching is a way of externalizing ideas, of turning internal thoughts public, of making fleeting thoughts more permanent [1]. People usually use sketches to express and record their ideas in many domains, including mechanical engineering, software design, information architecture [2] and map schematizing [3]. Although it may be easy for people to understand the designer's intention, which is presented in his sketches intangibly, it is not an easy case for computers. The ambiguity of sketches causes many problems in recognition.

The online graphics recognition problem can be specified into three levels: primitive shape recognition, composite object recognition, and document-level recognition and understanding [4]. For composite object recognition, existing approaches can be divided into two categories: the SPR (Statistical Pattern Recognition) approaches, such as Neural Networks (NN) [5] and Hidden Markov Model (HMM) [6], and the SSPR (Syntactical and Structural Pattern Recognition) approaches, such as Attributed Relation Graph (ARG) and Region Adjacent Graph (RAG) [7][8]. These methods mainly focus on unitary or binary relationships. In addition, contextual (top-down) knowledge has also been used to recognize freehand sketches of simple 2-D mechanical devices [9].

Sketches have the advantage of conveying elements and spatial relations in the real world with elements and spatial relations on paper and describing visuospatial ideas

directly [1]. Users' sketchy objects are composed of both primitive shapes (e.g. line segments, arc segments, and ellipses) and their topological spatial relationships. These relationships among the primitive shapes play an important role in identifying composite objects. For example, when people distinguish one object from the others, they usually pay more attention to the structure of its components than to the absolute position, size, and orientation of each component. Topological relationship is a particular subset of geometric relations that are invariant under geometric transformations such as translation, rotation, and scaling. Traditionally, topological relationship is defined between two objects, such as regions, lines or points [10]. Zhan [11] provided eight binary topological relations between two fuzzy regions that are distinguished by the 9-intersection model [10], e.g. disjoint, contains, inside, equal, meet, covers, coveredby, and overlap. However, binary topological relationship itself cannot carry all information of the internal structure of composite objects. For example, it is impossible to distinguish the relative sequence of three lines, which are parallel, only with binary topological relationships. However, with the proposed ternary relationships, we are able to distinguish it easily. In this paper, two ternary relations are introduced, which, along with certain binary relationships, are used to describe the internal structure of composite graphic objects. A constrained partial permutation algorithm is also presented to recognize sketchy objects using these relationships. Experimental results show that this approach is both efficient and effective for online composite graphics recognition in our sketch-based graphics input system – SmartSketchpad [12].

The remainder of this paper is organized as follows: Section 2 introduces 5 classes of binary topological relationships and two ternary topological relationships that are used in our system. Section 3 presents a constrained partial permutation algorithm for recognition of sketchy objects using their topological relationships. Experiments and evaluations are presented in Section 4. Section 5 presents concluding remarks.

2 Topological Spatial Relationship

2.1 Notations

The point in 2D plane can be denoted as Pt . The primitive shape in a graphic object, as we consider, can be line segment (denoted as Ls), arc segment (denoted as As), or ellipse (denoted as E). The primitive shape is denoted as P , and $CLASS(P)$ is the class of P . L is the line in which the segment (Ls) locates and Ra is the circle on which the arc segment (As) locates. The relationship between two primitive shapes (e.g. P_1 and P_2) can be denoted as $R(P_1, P_2)$ and $R'(P|P_1, P_2)$ is a ternary relationship among three primitive shapes (e.g. P, P_1 , and P_2). Moreover, we use $R(P, \bullet)$ or $R'(P|\bullet, \bullet)$ to express all binary or ternary relationships which contain P and $\Phi(R(P, \bullet))$ or $\Phi(R'(P|\bullet, \bullet))$ is the number of relationships in $R(P, \bullet)$ or $R'(P|\bullet, \bullet)$.

In addition, $f(x, y)$ is the function of the primitive shape. For line, $f(x, y) = ax + by + c$, where $b \geq 0$ and if $b = 0$, $a > 0$. For ellipse, $f(x, y) = (x - x_0)^2/a^2 + (y - y_0)^2/b^2 - c$, where $c > 0$ and $a, b \neq 0$. For circle, $f(x, y) = (x - x_0)^2 + (y - y_0)^2 - c$, where $c > 0$.

2.2 Binary Topological Relationships

We first define five binary spatial relationships that are useful for sketch recognition as follows.

Definition 1. Binary Topological Relationships

Given two primitive shapes P_1 and P_2 , the binary spatial relationship $R(P_1, P_2)$ can be defined as follows, which are also illustrated in Fig. 1.

- (i). **Interconnection** (R_{ic}): P_1 and P_2 have common end points, or two ellipses join together (Fig. 1(a)). Denoted as $R_{ic}(P_1, P_2)$.
- (ii). **Tangency** (R_t): The end points of P_1 are quite close to (or touching) some inner points of P_2 , or a line segment is tangent to an ellipse (Fig. 1(b)). Denoted as $R_t(P_1, P_2)$.
- (iii). **Intersection** (R_{is}): If P_1 and P_2 have common inner points (Fig. 1(c)), we define it as $R_{is}(P_1, P_2)$.
- (iv). **Parallelism** (R_p): P_1 and P_2 are line segments and are approximately parallel [13] within a sufficiently close distance (Fig. 1(d)). Denoted as $R_p(P_1, P_2)$.
- (v). **Concentric** (R_c): The centers of two ellipses or arc segments are sufficiently close (Fig. 1(e)). Denoted as $R_c(P_1, P_2)$.

According to above definition, the relationship R can be recognized even after geometric transformations. Therefore, these binary relationships are insensitive to the direction, the size and the rotation.

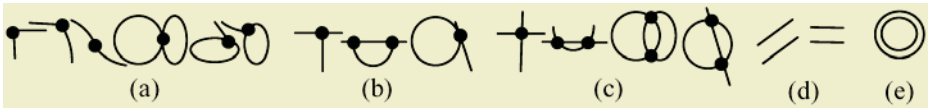


Fig. 1. Some examples of binary spatial relationship, with joint points drawn in thick black.

2.3 Ternary Topological Relationship

The above binary relationships are insufficient to distinguish all possible spatial relations among components. For instance, we cannot distinguish the relative sequence of three parallel lines or three concentric ellipses with only binary relationships. Therefore, the ternary relationship is introduced. Before we present the detailed definitions of the ternary relationship, we first define another two binary relationships: the *over* relationship and the *below* relationship. Given two primitive shapes P and P' , which are not exactly the same in their geometric functions, and if the function of P is $f(x, y)$, we define:

Definition 2. The over relationship

If $\forall Pt$ in the primitive shape P' , $f(Pt) \geq 0$ and there is at most one point Pt' at which $f(Pt') = 0$, we can say P' is over P , denoted as $P' \uparrow P$.

Definition 3. The below relationship

If $\forall Pt$ in the primitive shape P' , $f(Pt) \leq 0$ and there is at most one point Pt' at which $f(Pt') = 0$, we can say P' is below P , denoted as $P' \downarrow P$.

Although the over/below relationships for circles or ellipses are rotation-invariant, the two relationships for lines are sensitive to rotation. Therefore, we define the following two ternary relationships based on the over/below relationships.

Definition 4. The middle relationship

Given three primitive shapes (P, P_1, P_2) , if $P_1 \uparrow P$ and $P_2 \downarrow P$, we can say that P has the middle relationship with the other two primitive shapes $(P_1$ and $P_2)$, denoted as $R'_M(P|P_1, P_2)$.

Every three primitive shapes may have more than one (at most two) middle relationship. For example, the object in Fig. 2 has three line segments and one ellipse. The three line segments have two middle ternary relationships: $R'_M(P_a|P_c, P_b)$ and $R'_M(P_b|P_a, P_c)$ which are drawn from the over/below relationships $P_c \uparrow P_a$, $P_c \downarrow P_b$, $P_a \uparrow P_b$ and $P_b \downarrow P_a$.

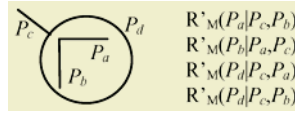


Fig. 2. An example object and its ternary relationships.

Definition 5. The parallel-middle relationship

If $R_p(P_1, P_2)$ or $R_c(P_1, P_2)$, and there is a primitive shape P , which can meet the conditions: $P \downarrow P_1$ and $P \uparrow P_2$, we say that these three primitive shapes have the parallel-middle relationship, denoted as $R'_{p-M}(P|P_1, P_2)$.

Apparently, the parallel-middle relationship can meet the rotation-invariance, that is, $R'_{p-M}(P|P_1, P_2) = R'_{p-M}(P|P_2, P_1)$. In addition, for the line segments, this ternary relationship is sensitive to the parallel relationship. Therefore, we cannot use the traditional parallel definition. For example, P_c and P_a in Fig. 3 may be considered as parallel under the traditional definition. Therefore, we use Revankar and Yegnanarayana's method [13] to judge the parallel relationship.

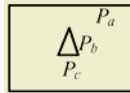


Fig. 3. The parallel-middle relationship is sensitive to the definition of the parallelism.

3 Sketch Recognition

The key problem in the composite object recognition process is how to match a sketchy object (SO) with a pre-defined object (PO). Obviously, enumeration of all possible

cases is equivalent to a Partial Permutation problem, which is proved to be an NP-complete problem [14]. For an object composed of more than 8 components, it is not suitable to enumerate all cases in on-line recognition system. In fact, most permutations are illegal mappings due to violation of some constraints and can be neglected directly during the permutation process. Hence, this partial permutation is a constrained one and is referred to as the *Constrained Partial Permutation* by Xu et al. [15]. Xu et al.'s method [15] is based on binary relationships. In this paper, we improve it by handling with the ternary relationships as well as binary relationships.

3.1 Constrained Partial Permutation

Definition 6. Constrained Partial Permutation

Given that SO contains m primitive shapes (e.g. O_1, O_2, \dots, O_m) and PO contains n primitive shapes (e.g. P_1, P_2, \dots, P_n) where $m \leq n$, select m integers from $[1..n]$, and then rank them in a list as enumeration sequence, written as $B(1)B(2)\dots B(m)$, where, $1, 2, \dots, k, \dots, m$ are the positions in the list and $B(1), B(2), \dots, B(k), \dots, B(m)$ are the values ($1..n$) at these positions in the list. We regard SO as a part of PO, only if we can find a possible enumeration sequence which satisfies all of the following three constraints:

Constraint 1: $\forall r \in [1..m]$, $\text{CLASS}(O_r) = \text{CLASS}(P_{B(r)})$, $\Phi(R_{\text{IC,T,IS,P,or C}}(P_{B(r)}, \bullet)) \geq \Phi(R_{\text{IC,T,IS,P,or C}}(O_r, \bullet))$, $\Phi(R'_{\text{M}}(P_{B(r)} | \bullet, \bullet)) \geq \Phi(R'_{\text{M}}(O_r | \bullet, \bullet))$ and $\Phi(R'_{\text{P-M}}(P_{B(r)} | \bullet, \bullet)) \geq \Phi(R'_{\text{P-M}}(O_r | \bullet, \bullet))$

Constraint 2: $\forall r, s \in [1..m]$, if there is a binary relationship $R_{\text{IC,T,IS,P,or C}}(O_r, O_s)$, there must be a relationship $R_{\text{IC,T,IS,P,or C}}(P_{B(r)}, P_{B(s)})$ of the same type.

Constraint 3: $\forall r, s, t \in [1..m]$, if there is a ternary relationship $R'_{\text{M or P-M}}(O_r | O_s, O_t)$, there must be a relationship $R'_{\text{M or P-M}}(P_{B(r)} | P_{B(s)}, P_{B(t)})$ of the same type.

To improve the efficiency of constrained partial permutation, we define four denying rules as follows:

Definition 7. Single Denying Rule:

Given $r \in [1..m]$ in SO and $i \in [1..n]$ in PO, P_i cannot be put into the place r , that is, $B_r \neq i$, written in a 2-tuple (i, r) , if one of the following conditions can be met: (1) $\text{CLASS}(O_r) \neq \text{CLASS}(P_i)$; (2) $\Phi(R_{\text{IC,T,IS,P,or C}}(P_i, \bullet)) < \Phi(R_{\text{IC,T,IS,P,or C}}(O_r, \bullet))$; (3) $\Phi(R'_{\text{M}}(P_i | \bullet, \bullet)) < \Phi(R'_{\text{M}}(O_r | \bullet, \bullet))$; (4) $\Phi(R'_{\text{P-M}}(P_i | \bullet, \bullet)) < \Phi(R'_{\text{P-M}}(O_r | \bullet, \bullet))$.

Definition 8. Pair Denying Rule 1:

Given $r, s \in [1..m]$ in SO and $i, j \in [1..n]$ in PO, and P_j has been put into the place s (that is, $B_s = j$), if O_r has the binary relationship $R_{\text{IC,T,IS,P,or C}}(O_s, O_r)$ but P_i does not have the same binary relationship $R_{\text{IC,T,IS,P,or C}}(P_i, P_j)$, P_i cannot be put into the place r , that is, $(i, r) - (j, s)$.

Definition 9. Pair Denying Rule 2:

Given $r, s \in [1..m]$ in SO and $i, j \in [1..n]$ in PO, and P_j has been put into the place s (that is, $B_s = j$), if O_r has the ternary relationship $R'_{\text{M, or P-M}}(O_r | O_s, \bullet)$ or $R'_{\text{M, or P-M}}(O_s | O_r, \bullet)$ but P_i does not have the same ternary relationship $R'_{\text{M, or P-M}}(P_i | P_j, \bullet)$ or $R'_{\text{M, or P-M}}(P_j | P_i, \bullet)$, P_i cannot be put into the place r , that is, $(i, r) - (j, s)$.

Definition 10. Triangle Denying Rule:

Given $r, s, t \in [1..m]$ in SO and $i, j, k \in [1..n]$ in PO, P_j has been put into the place s and P_k has been put into the place t (that is, $B_s=j$ and $B_t=k$), if O_r has the ternary relationship $R'_{M, \text{ or } P-M}(O_r|O_s, O_k)$ or $R'_{M, \text{ or } P-M}(O_s|O_r, O_k)$ but P_i does not have the same ternary relationship $R'_{M, \text{ or } P-M}(P_i|P_j, P_k)$ or $R'_{M, \text{ or } P-M}(P_j|P_i, P_k)$, P_i cannot be put into the place r , that is $(i, r)-(j, s)(t, k)$.

Therefore, given a permutation $B(1)B(2)...B(m)$, we can derive a function $f: [1..m] \rightarrow [1..n]$, where $f(i)=B(i)$ for any $i \in [1..m]$. We obtain four denying rules according to the following steps:

- Step 1: if f is rejected by constraint 1, there must exist $r (r \in [1..m])$, which let O_r and $P_{B(r)}$ meet the single denying rule. Then we obtain a single denying rule $(B(r), r)$.
- Step 2: if f is rejected by constraint 2, there must exist $r, s \in [1..m]$, which let O_r, O_s and $P_{B(r)}, P_{B(s)}$ meet the pair denying rule 1. Then we obtain a pair denying rule $(B(r), r)-(B(s), s)$.
- Step 3: if f is rejected by constraint 3, there must exist $r, s \in [1..m]$, which let O_r, O_s and $P_{B(r)}, P_{B(s)}$ meet the pair denying rule 2, or there must exist $r, s, t \in [1..m]$, which let O_r, O_s, O_t and $P_{B(r)}, P_{B(s)}, P_{B(t)}$ meet the triangle denying rule. Then we obtain a pair denying rule $(B(r), r)-(B(s), s)$ or a triangle denying rule $(B(r), r)-(B(s), s)(B(t), t)$.

In order to acquire all possible permutations, we enumerate all such m -digit integers from the smallest, i.e., $123...m$, to the largest, i.e., $n(n-1)(n-2) \dots (n-m+1)$. Each time we give its next permutation by finding the smallest m -digit number that is not found before. The denying rules will be generated gradually. If the current permutation is $b_1b_2...b_{k-1}ib_{k+1}...b_m$, and it is rejected by the single denying rule (i, k) , we directly skip all permutations with the form $b_1b_2...b_{k-1}iB_{k+1}...B_m$, where $B_x (x=k+1..m)$ belongs to the set $\{1, 2, 3, \dots, n\} - \{b_1, b_2, \dots, b_{k-1}, i\}$ and $B_p \neq B_q (p, q=k+1..m)$. If the current permutation is $b_1b_2...b_{k-1}ib_{k+1}...b_{t-1}jb_{t+1}...b_m$, where $t > k$, and it is same when it is rejected by the pair denying rule $(i, k)-(j, t)$ or the triangle denying rule $(i, r)-(j, s)(t, k)$. By doing so, many illegal mappings (permutations) can be pruned directly in the enumerating process.

3.2 Similarity

The similarity between SO and PO in a given match (legal mapping/permutation $B(1)B(2)...B(m)$) is denoted by $\text{Sim}_B(\text{SO}, \text{PO})$ and defined as the weighted sum of the similarities of all pairs of matched primitives, as follows. $L(P)$ is the length (for line/arc segments) or perimeters (for ellipses) of a primitive shape P .

$$\text{Sim}_B(\text{SO}, \text{PO}) = \sum_{r=1}^m \omega_r \frac{\min(L(P_{B(r)}), L(O_r))}{\max(L(P_{B(r)}), L(O_r))} \quad \text{where} \quad \omega_r = L(O_r) / \sum_{j=1}^m L(O_j), r \in [1..m], \quad (1)$$

Then, the final similarity between SO and PO is defined as the maximal similarity under all possible mappings/permutations, as follows.

$$\text{Sim}(\text{SO}, \text{PO}) = \max_{B \in \Psi(\text{SO}, \text{PO})} \text{Sim}_B(\text{SO}, \text{PO}) \quad (2)$$

where $\Psi(\text{SO}, \text{PO})$ is the set of all legal mappings between SO and PO.

4 Experiments

4.1 Experimental Environments

In this experiment, we simulate the sketch-based graphics input and recognition process in practical applications. Firstly, 349 composite graphic objects are created to form a database. Each object in the database contains no more than 15 components. Secondly, we use the method of Xu et al. [15] to generate queries from these objects randomly and match these queries with those objects in the database. For a given graphic object, which has m components, denote the width and height of the object as w and h respectively. A query is generated according to a noise rate τ_n , which is used to simulate the drawing noises, and a completion rate τ_c , which is used to simulate the incomplete form.

Algorithm 2: Query Generating. (Given a graphic object with n components and its completion rate τ_c)

Step 1: randomly select $n * \tau_c$ components.

Step 2: generate simulation noises for each component.

Step 2.1: generate a horizontal shifting factor ι_H and a vertical shifting factor ι_V between $-\tau_n$ and $+\tau_n$. Then shift this component by $\iota_H * w$ horizontally and by $\iota_V * h$ vertically.

Step 2.2: generate a random scaling factor ι_s between $1 - \tau_n$ and $1 + \tau_n$, and then rescale the component according to ι_s .

Step 2.3: for each component, generate a random rotation factor ι_R between $-\tau_n * \pi$ and $\tau_n * \pi$, and then rotate the component according to ι_R counterclockwise.

Step 3: Combine these $n * \tau_c$ components as a whole group, then shift, rotate, and rescale this group, randomly, to form a query q .

Fig. 4(a) shows a regular object stored in the database and Fig. 4(b) is the generated query object for this regular object at $\tau_c = 0.9$ and $\tau_n = 0.1$.

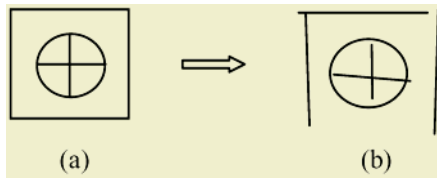


Fig. 4. An example of generated query object from a regular object. (a) The original model object stored in the database; (b) the generated query at $\tau_c = 0.9$ and $\tau_n = 0.1$.

For the i -th object in the database, we can generate a query q_i randomly according to τ_n and τ_c . Next, we rank all POs in the database according to their similarities to q_i in a descending order. Denote the position of the i -th object itself in the ranking list as $Rank_i$, e.g., $Rank_i$ equals to 2, that is, the i -th object has the second highest similarity to the query q_i . The recall rate R_n is defined as:

$$R_n = \sum_{i=1}^{Total_Query} \frac{RANK_{\leq n}(q_i)}{Total_Query}, \text{ where } RANK_{\leq n}(q_i) = \begin{cases} 1 & \text{if } (Rank_i \leq n) \\ 0 & \text{if } (Rank_i > n) \end{cases} \quad (3)$$

The experiment environment is Pentium III 1.3G CPU, 256MB memory, Windows XP, Visual C++ 6.0.

4.2 Performance Evaluation

In the experiment, τ_n is set to be 0, 0.1, and 0.2, respectively, to simulate different drawing styles. $\tau_n=0$ is set to simulate a very formal drawing, which has few noise. $\tau_n=0.1$ is set to simulate an ordinary user-sketched drawing, which has some noises. $\tau_n=0.2$ is set to simulate a haste drawing, which has many noises when user draws freely. On the other hand, τ_c increases from 0.5 to 1.0 simulating different completion status. R_i is defined in Eq (3). R_i in percentage under different τ_n and τ_c are shown in Fig. 5.

From Fig. 5 we can see that our algorithm is relatively sensitive to noises when a few components have been drawn. As more components have been drawn (i.e., as the object is more complete), higher recall rates can be achieved. We can also see that, when noises are relatively small ($\tau_n=0$ or 0.1), the sketchy graphic object can be successfully recognized before it has been drawn completely. For instance, when a user has drawn 80% of his/her intended object with $\tau_n=0.1$, the rate of successful recognition is above 80%. Hence, we can draw the conclusion that our approach can achieve good performance under noises for incomplete sketchy input.

Our algorithm uses not only the binary relationship but also the ternary relationship to recognize composite objects. This method can get a better performance than the method only using the binary relationships (that is, Xu et al.'s method [15]). Fig. 6 shows that the recall ratio of our algorithm is higher than Xu et al.'s algorithm when $\tau_n=0.1$ and $\tau_n=0.2$.

The time cost in the recognition process is also a much-concerned factor for evaluating our approach's performance. Especially, in our real-time interactive sketch recognition environment, the time cost should be as small as possible. The time cost comparison between our approach and Xu et al.'s approach is listed in Table 1. From this table, we can find that our algorithm is more efficient than Xu et al.'s algorithm. From all comparisons in our experiment, we find that our approach can save 56.3% time compared to Xu et al.'s algorithm in average and the recall ratio is also higher than Xu et al.'s approach.

Table 1. Comparison of time cost of our algorithm and Xu et al.'s algorithm for some shapes (Milliseconds).

The number of components in the object	8	10	12	14
Time cost of Xu et al.'s algorithm	30	310	100	750
Time cost of our algorithm	10	50	70	400

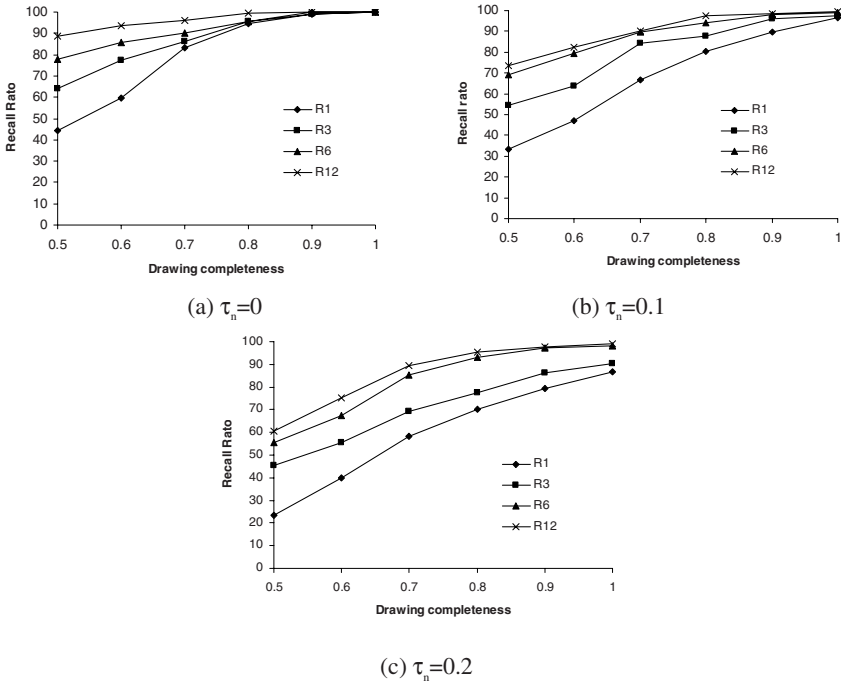


Fig. 5. Performance evaluation of the composite graphic object recognition.

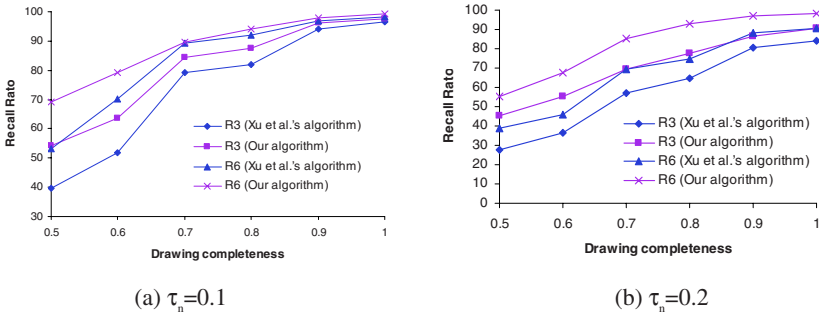


Fig. 6. Comparison of recall ratio between our algorithm and Xu et al.'s algorithm [15].

5 Conclusion

In this paper, we have proposed two ternary spatial relationships among graphic components. The constrained partial permutation algorithm [15] has been improved to handle both binary and ternary topological spatial relationships for recognition of sketchy objects input by users in an online manner. From the experimental results, we can see that our improvement is more efficient and effective than Xu et al.'s method [15] and is practical in real-time sketch-based graphics input and recognition systems.

Acknowledgement

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 1073/02E].

References

1. Tversky, B.: What do Sketches Say about Thinking? In: Proc. of AAAI Spring Symposium-Sketch Understanding. (2002) 148-151
2. Alvarado, C., Oltmans, M., Davis, R.: A Framework for Multi-Domain Sketch Recognition. In: Proc. of AAAI Spring Symposium-Sketch Understanding. (2002) 1-8
3. Skubic, M., Blisard, S., Carle, A., Matsakis, P.: Hand-Drawn Maps for Robot Navigation. In: Proc. of AAAI Spring Symposium-Sketch Understanding (2002) 140-147
4. Liu, W.: On-Line Graphics Recognition. To appear in Post-Proc. of GREC2003 (LNCS 2004)
5. Lee, S.W., Kim, Y.J.: A New Type of Recurrent Neural Network for Handwritten Character Recognition. In: Proc. of ICDAR95. Vol. 1 (1995) 38-42
6. Bicego, M., Murino, V.: 2D Shape Recognition by Hidden Markov Models. In: Proc. of 11th International Conference on Image Analysis and Processing. (1991) 20-25
7. Lladós, J., Martí, E., José, J.: Symbol Recognition by Error-Tolerant Subgraph Matching Between Region Adjacency Graphs. IEEE Trans. on PAMI. 23(10) (2001) 1137-1143
8. Xu, X., Liu, W., Jin, X., Sun Z.: Sketch-Based User Interface for Creative Tasks. In: Proc. of 5th Asia Pacific Conference on Computer Human Interaction. (2002) 560-570
9. Alvarado, C., Davis, R.: Resolving Ambiguities to Create a Natural Sketch Based Interface. In: Proc. of IJCAI-2001. (2001)
10. Egenhofer, M.J., Franzosa, R.: Point-Set Topological Spatial Relations. International Journal of Geographical Information Systems. 5(2) (1991) 161-174
11. Zhan, F.B.: Topological Relations Between Fuzzy Regions. In: Proc. of the 1997 ACM Symposium on Applied Computing. (1997) 192-196
12. Liu, W., Jin X., Sun, Z.: Sketch-Based User Interface for Inputting Graphic Objects on Small Screen Devices. Lecture Notes in Computer Science 2390 (2002) 67-80
13. Revankar, S., Yegnanarayana, B.: Machine Recognition and Correction of Freehand Geometric Line Sketches. In: Proc. of IEEE International Conference on Systems, Man, and Cybernetics. (1991) 87-92
14. Mehlhorn, K.: Graph Algorithm and NP-Completeness, Springer-Verlag. (1984)
15. Xu, X., Sun, Z., Peng, B., Liu, W.: An Online Composite Graphics Recognition Approach Based on Matching of Spatial Relation Graphs. To appear in International Journal on Document Analysis and Recognition.