IJDAR (2004) 7: 44–55
Digital Object Identifier (DOI) 10.1007/s10032-004-0126-3

International Journal on Document Analysis and Recognition
IJDAR

An online composite graphics recognition approach based on matching of spatial relation graphs

Xu Xiaogang[1,2], Sun Zhengxing[1,2], Peng Binbin[1,2], Jin Xiangyu[1,2], Liu Wenyin[3]

[1] State Key Lab for Novel Software Technology, Nanjing University, Nanjing 210093, P.R. China
e-mail: {xuxg,szx,pbb,jxy}@graphics.nju.edu.cn
[2] Department of Computer Science and Technology, Nanjing University, Nanjing 210093, P.R. China
[3] Department of Computer Science, City University of Hong Kong, Hong Kong SAR, P.R. China
e-mail: csliuwy@cityu.edu.hk

Received: March 13, 2003 / Accepted: March 13, 2004
Published online: June 1, 2004 –

Abstract. A spatial relation graph (SRG) and its partial matching method are proposed for online composite graphics representation and recognition. The SRG-based approach emphasizes three characteristics of online graphics recognition: partial, structural, and independent of stroke order and stroke number. A constrained partial permutation strategy is also proposed to reduce the computational cost of matching two SRGs, which is originally an NP-complete problem as is graph isomorphism. Experimental results show that our proposed SRG-based approach is both efficient and effective for online composite graphics recognition in our sketch-based graphics input system – SmartSketchpad.

Keywords: Graph matching – Spatial relation graph – Sketch-based user interface – Online graphics recognition

1 Introduction

Nowadays computers are expected not only to perform computational tasks, but also to assist people in creative tasks, including drawing, writing, designing, and so on. This interests more and more human–computer interaction (HCI) researchers to investigate techniques that can strengthen computers with humanlike properties such as those allowing for quick drafting, creative designing, informal communication, ambiguous expression, and instant feedback from computers to users. That is, computers are required to bend to human-preferred interaction mode, not the other way around [16]. The essence of the idea is to make computers more intelligent, more convenient to use, and more adaptable to the human-preferred communication mode.

In more and more situations, people are accustomed to writing down their improvisatory ideas freely by drawing graphic objects. For them, the ability to rapidly deliver their ideas using graphic objects with uncertain types, indefinite sizes, irregular shapes, and inaccurate positions is most important to the note-taking process. These ambiguities encourage authors to explore more ideas without being bogged down with unnecessary details, e.g., colors, fonts, and precise alignments. The traditional way to perform these tasks is by sketching on paper with a pen, which is a preferred choice for creative brainstorming [11,15]. However, most current computer-aided drafting tools, including Microsoft Office, PhotoDraw, Visio, and AutoCAD, discard the traditional pen-based design interface and require users to select graphic patterns from lots of toolbar buttons or menu items. This is a computer-oriented interaction style. Users frequently find it inconvenient due to the numerous mouse clicks. They also complain that they have to memorize the precise position of each toolbar button or menu item since they cannot focus on the design idea itself when utilizing these tools to deliver their bursting creative ideas. Therefore, they cannot finish the design in continuous steps due to too many interruptions. Moreover, it is unrealistic to do such design work on small screen devices since there is not enough room to accommodate so many toolbar buttons or menu items on their small screens.

Sketching with a pen, an informal interaction mode, which has been shown especially valuable for creative design tasks [10], is the very solution to the situation mentioned above. Furthermore, it would be more helpful if the sketchy shape could be recognized and converted into the user-intended regular shape immediately. Though the sketchy shape in its rough state contains more information than the regularized one, the regularized shape is better for users to communicate and to recall their original intention of this sketch. The process for sketchy shape input and instant recognition is referred to as online graphics recognition [13,20,21]. The key idea behind online graphics recognition in our research is to input regular shapes (e.g., ellipses, circles, rectangles, triangles, arrowheads, straight lines, etc.) or composite graphic objects (e.g., iconic symbols of hard disks, computers, switches, etc.) by quickly sketching their approx-

imate line shapes with an electronic pen on tablets, just like writing with a pen on paper.

When drawing a graphic object, a user tends to divide it into several primitive shapes and draw them one by one. Therefore, the entire online graphics recognition process is divided into two phases: primitive shape recognition/regularization and composite graphic object recognition [22]. In the primitive shape recognition/regularization phase, we first find out the potential primitive shape among the sketchy strokes a user draws, then recognize and regularize this primitive shape, and finally show the regularized drawing to the user immediately. This immediate feedback strategy makes the user interaction smoother and more natural. An extra advantage of this strategy is that it can reduce intrastroke and interstroke noise, which is usually introduced by the user's limited capability or low professional ability. The techniques used in this phase have been reported by Liu et al. [21], Jin et al. [13], and Sun et al. [35]. In the composite graphic object recognition phase, the recognized and regularized primitive shapes that belong to the same composite graphic object are grouped together according to the rule of proximity. Similarities are calculated between the user-drawn graphic object and the candidate ones in the database. The graphic objects that are the most similar to what the user has drawn are suggested to the user. A complex and ad hoc scheme to compare the similarity between two composite graphic objects has been presented by Liu et al. [22].

In this paper, we focus on the problem of the composite graphic object recognition. We propose a spatial relation graph (SRG) to represent the spatial relationship among the primitives in a composite graphic object. The similarity between two composite graphic objects is then assessed by matching their SRGs. A partial similarity measurement is also put forward for SRG matching. To reduce the computational cost of matching two SRGs, which is an NP-complete problem like graph isomorphism [24], we also propose a constrained partial permutation strategy to prune many illegal mappings/matchings. The proposed scheme has been implemented in our SmartSketchpad system [21,22,35]. Experimental results and evaluations have shown its efficiency and effectiveness.

The rest of this article is organized as follows. Some related work is presented in Sect. 2. In Sect. 3, we describe some characteristics of the online graphics input process that should be considered when performing the online graphics recognition task. In Sect. 4, we present our proposed spatial relation graph (SRG) and its application in composite graphics recognition. We also give a brief introduction to the constrained partial permutation algorithm we use to reduce the computational cost of SRG matching. Experimental results and evaluations are described in Sect. 5. Finally, we present our concluding remarks in Sect. 6.

2 Related work

An intuitional approach to graphics recognition is to build a decision tree in which each leaf represents a class of graphic objects and each edge represents a classifying rule. The composite graphic object is recognized according to these rules descending from the root of the decision tree. However, this approach suffers from its nonextensibility. When a new class of graphic objects is added, a specific classifying rule must be added and the existing rules must be modified. This approach has been revived recently with the incorporation of fuzzy rules and visual language grammars that use statistical information on the occurrence frequency of particular features, as reported by Fonseca's group [7].

Some existing approaches are based on statistical machine learning algorithms, such as neural networks (NN) [18] and support vector machine (SVM) [28,31]. The advantage of these approaches is that they are robust to noise and the system can be easily extended. However, the critical point of these approaches is extracting the features without information distortion. A variety of approaches have been proposed to solve this problem, mainly converging into two categories: pixel-based and stroke-based [14]. Another point is organizing the classifiers efficiently, which are usually constructed by several subclassifiers. In addition, the hidden Markov model (HMM), which is widely used in speech recognition [32], is introduced to recognize composite graphic objects. The essence of this approach is to determine the posteriori probability for a class given an observed sequence where the jump from one state to another is described by a Markov process. Recent developments incorporate HMM into 2D-sketch recognition [27]. Another new trend is to use hybrid NN/HMM approaches. The fatal disadvantage of this approach is that a large human-created training set is required.

Graphs are powerful data structures for relational descriptions in structural pattern recognition. By assigning suitable meanings to nodes and edges of graphs, it is possible to achieve complete and univocal representations of objects [6]. Typically, nodes represent the parts of an object and edges represent relations between the parts. Graphs also have many important properties, such as being translation invariant, rotation invariant, scale invariant, and transform invariant. Attributed relation graph (ARG) [19,25], 2D strings [3], and region adjacent graph (RAG) [23] are typical approaches to representing spatial content. A 2D string is an arraylike representation of the graphic primitives of an object in a scan sequence from left to right and from bottom to top. It has been used in content-based image retrieval based on spatial similarity. The 2D string is a good representation of certain types of spatial relations between nonoverlapping objects [29]. Lee and Hsu [17] also proposed 2D C-strings to process the spatial relations between overlapping objects. ARG represents individual objects or graphic primitives by graph nodes and their relations by arcs between such nodes. ARG is the most general representation method for spatial relations, but matching between ARGs also has exponential computational

complexity. Approximate ARG matching methods with lower computational complexity are presented by Almohamad and Duffuaa [1], Christmas et al. [4], Ranganath and Chipman [33], and Gold and Rangarajan [9]. But their main disadvantage is that they may get trapped in local minima and miss the optimal solution [29]. A recent contribution to matching of ARGs is proposed by Messmer and Bunke [26]; it has the advantage of avoiding the matching of the query with every stored model and an expensive preprocessing step for building an ARG index structure [29]. Though much work has been done on ARG, a critical drawback is its sensibility to noise and distortion. Therefore, Lladós et al. [23] proposed RAG, in which graph nodes represent the minimal closed regions, to allow for matching between noise-polluted or distorted graphs with limited regions. However, since regions are the minimal processing units, it cannot process the inner structure of a specific region. Moreover, if there is no closed region, RAG will not work properly.

Contextual (top-down) knowledge has been used by Alvarado et al. [2]to recognize freehand sketches of simple 2D mechanical devices. They built a prototype system that interpreted and understood a user's sketch as it was being drawn [5]. Hammond and Davis [12] built a multilayer framework, including preprocessing, selection, recognition, and identification, to recognize multistroke objects in Unified Modeling Language (UML) by their geometrical properties. In CALI, Fonseca et al. [8] used temporal adjacency to recognize a user's sketchy shapes, such as triangles, lines, rectangles, circles, diamonds, and ellipses, using multiple strokes. In addition, they further extended this approach to identifying useful shapes such as arrows, crossing lines, and unistroke gesture commands. Saund [34] provided the computer-vision approaches to finding perceptually salient, compact closed-region structures in hand-drawn sketches and line art.

3 Online graphics inputting process

The online graphics inputting process is a real-time interaction between user and machine. Making the inputting process more intelligent, natural, and convenient to use is the ultimate goal of online graphics recognition. When performing the online graphics recognition task, some important characteristics should be considered.

3.1 Partial

When drawing a graphics object, a user tends to divide it into several primitive shapes and draw them one by one. With the completion of the drawing, the user's intention becomes increasingly clearer and the recognition precision increases. To save the user from expending effort in drawing, it is important for the online graphics recognition process to predict the user's intention before the sketchy drawing is completed. The earlier the prediction starts, the more effort it saves. In this case, there is an underlying assumption that if one candidate object is composed of fewer primitive shapes than the current incomplete drawing, this object cannot be the user's intention. The similarity calculation for real-time prediction is asymmetric between the incomplete object and other candidate objects. That is, if the incomplete object is part of a candidate object, they are considered highly similar because the incomplete object can be completed later, whereas if the incomplete object contains certain components that do not exist in the candidate object or the incomplete object contains more components than the candidate object, the candidate object is considered not to conform to the user's intention and the similarity should be very low, no matter how similar the corresponding parts are. Moreover, if the incomplete object is part of two or more candidate objects, the candidate object with the fewest components will have the highest similarity.

3.2 Structural

Many current shape recognition strategies, which are inherited from image processing and image understanding areas, regard the contour of objects or the area that are surrounded by the contour as the objects' major features and discard the inner structure within the contour of objects. However, the inner structure usually plays a more important role in identifying graphic shapes than the contour and the area. We believe that a reasonable graphics recognition strategy should consider the object structure as one of the most distinguishable features of objects and that a reasonable graphics recognition strategy should not only satisfy the three traditional invariabilities, i.e., translation invariant, rotation invariant, and scale invariant, but also satisfy the *structure invariant* (*topology invariant*). Here structure invariant (topology invariant) means that when recognizing graphics we do not pay much attention to each component's absolute position, size, and orientation but focus on their general attributes, such as line types, and their relative spatial and scale relations.

3.3 Stroke-number and stroke-order free

An intuitive approach for graphics recognition is to compare the strokes of the source sketchy object (the one being recognized) with those of the candidate regular object one by one in pairs. This approach is applicable for the applications that have both a fixed number of strokes and a fixed stroke-drawing order, e.g., signature verification [30]. However, different users may have different opinions in decomposing a composite object into stroke combinations and input them in different orders. Even the same user may change his drawing style from time to time. This diversity makes it very difficult for us to compare stroke pairs of graphic objects in the same order. Moreover, it is extremely difficult to enumerate all possible sketching orders in both spatial and temporal dimensions. Hence, a good graphics recognition approach should be compatible with different stroke numbers and stroke-drawing orders.

4 Partial structural similarity calculation based on matching of spatial relation graphs

We have designed a special graph, called a spatial relation graph (SRG), for representing the sketchy graphic object drawn by users with a digital pen and the regular graphic objects predefined in the database. First, we identify their primitive components (e.g., lines, arcs, and ellipses) by breaking down multistroke into single-stroke pairs, connecting a chain of consecutive line segments or arc segments with joint endpoints into a longer straight line segment or a bigger arc, and merging overlapped line segments or arc segments into a single straight line or arc. Second, the latent spatial relations between the primitive component pairs are discovered. Finally, we obtain a SRG to represent the graphic object with its nodes representing the primitive components of the graphic object and its edges representing the spatial relations between the components.

After graphic objects are represented by SRGs, the problem of measuring the similarity of graphic objects becomes the problem of calculating the similarity of SRGs. If the source sketchy object (SSO) drawn by the user is part of the candidate regular object (CRO) predefined in the database, there must exist a legal mapping from the nodes of the source SRG to those of the candidate SRG. A legal mapping means a mapping under which each primitive component of the SSO has the same properties as its corresponding primitive component in the CRO and the spatial relationship between every two primitives of the SSO is the same as that between their corresponding primitives of the CRO. We calculate the similarity of the two graphic objects based on the similarity between the matched nodes and edges of their SRGs. Then we choose the maximal similarity under all legal mappings as the similarity between the SSO and the CRO.

In the rest of this section, we first introduce several representative spatial relations between the primitives and then give a formal definition to SRG of a graphic object in detail. Finally, we describe a similarity metric used in our SmartSketchpad system.

4.1 Spatial relations of graphic primitives

The primitive component in a graphic object can be line segment, arc segment, or ellipse. We denote the type set as Σ_T=$\{T_{Line}, T_{Arc}, T_{Ellipse}\}$. Given two graphic primitives P_1 and P_2, the spatial relation R between them can be expressed as an ordered pair P_1RP_2. We first introduce spatial adjacency relation (SAR), which is a relation of spatial proximity between two objects. We define five typical SARs, which are also illustrated in Fig. 1.

(i). Interconnection (R_{IC}): P_1 and P_2 have a common endpoint, or two ellipses joined together (Fig. 1a). An interconnection relation is symmetric.

(ii). Tangency (R_T): The endpoints 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. 1b). A tangency relation is asymmetric.

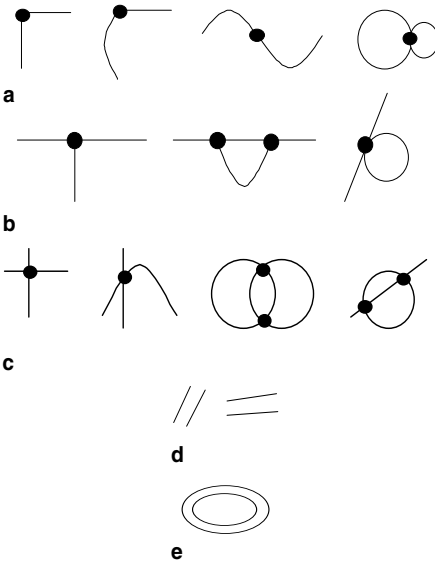


Fig. 1a–e. Examples of spatial adjacency relations, with joint points drawn in *thick black*

(iii). Intersection (R_{IS}): If P_1 and P_2have common inner points (Fig. 1c), we define $P_1R_{IS}P_2$. An intersection relation is symmetric.

(iv). Parallelism (R_P): P_1 and P_2 are line segments and are approximately parallel within a sufficiently close distance (Fig. 1d). A parallelism relation is symmetric.

(v). Concentric (R_C): The centers of two ellipses/circles are sufficiently close (Fig. 1e). A concentric relation is symmetric.

We denote the five representative SARs as a set $\Sigma R = \{R_{IC}, R_T, R_{IS}, R_P, R_C\}$. As shown in Fig. 1, SARs are insensitive to rotation.

However, the SARs are not sufficient to distinguish the possible spatial relations among graphic primitives. For instance, we cannot distinguish the relative sequence of the three parallel lines in Fig. 2a and the inner ellipse in Fig. 2b by using SAR only. Also, we cannot tell the tangency relation in Fig. 2c from that in Fig. 2d since they have the same type of SAR, but with different relative directions. Hence, we introduce relative position relations (RPR) to solve this problem. To simplify this problem, we define some reference directions such as horizontal, vertical, and two diagonal directions. According to these directions, we can determine the "before" relation (a specific RPR) of parallelism and different tangency relations. If P_1 appears before P_2 along the reference direction, we will denote this specific RPR as $P_1R_{BEF}P_2$. RPRs depend on the reference direction and are sensitive to rotation. Using R_{BEF}, we can easily solve the above-mentioned problem.

One thing that should be emphasized is that the thresholds to judge SARs are set differently for the SSO and the CRO. Due to the imprecision of user inputs of the SSO, sometimes the adjacency relation is very

ambiguous to judge. Figure 3 shows such an example. The user-intended relation between the line segments is interconnection. However, the user draws the lines imprecisely and the two line segments intersect at angles. Thus the relation between these two graphic primitives will be misjudged as an intersection and hence cannot be matched to the correct CRO. One solution is to make the relations not mutually exclusive. We set the thresholds loosely so that the relations could be easily met. Thus there will be several relations forming a relation set for an ambiguous situation. If the actual relation obtained from the CRO is a member of the relation set, we regard that mapping as successful. The SARs among the graphic primitives of the CRO are still mutually exclusive and they are judged by strict conditions since they are created formally with less noise.

4.2 Spatial relation graphs and their matching

The SRG of an object is a graph with its nodes representing the graphic primitives of this object and its edges representing the spatial relations between these primitives. Based on this representation, matching of two objects is done by matching their SRGs. Two SRGs are considered matched (as a perfect or less perfect match) if all or most of their corresponding nodes are matched (of the same type of primitive object) and all corresponding edges are matched (of the same type of spatial relation). As mentioned above, matching of two graphs is a highly complex process. To reduce the computational cost of matching two SRGs, we propose a constrained partial permutation strategy in which certain denying constraints are used to prune many illegal mappings (between corresponding nodes and edges) earlier in the enumeration process. Next, we first introduce some definitions and then present this strategy in detail.

Definition 1. Spatial relation graph (SRG): A SRG is defined as a 7-tuple $G = (V, E, T_A, T_E, E', G_E, P_E)$, where

(i). V is the set of graph nodes.

(ii). $E \subseteq V \times V$ is the set of graph edges.

(iii). $T_A : V \rightarrow A_V$ is a function assigning attributes to the nodes, where A_V is an attribute set including the graphic primitive type set Σ_T, the degree of each node, etc.

(iv). $T_E : E \rightarrow \Sigma R/2^{\Sigma R}$ is a function assigning labels to the edges, where ΣR is the symbolic label set of SARs

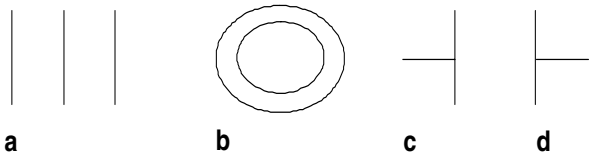


Fig. 2a–d. Examples of spatial adjacency relation that we cannot deal with

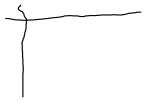


Fig. 3. An ambiguous relation: interconnection or intersection?

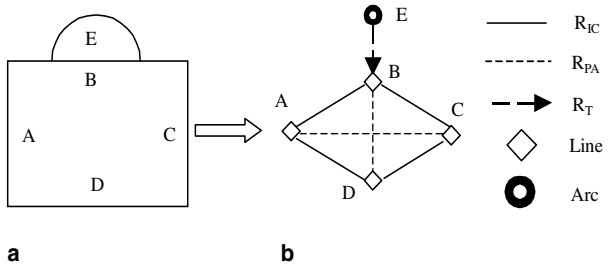


Fig. 4. a An exemple graphic object with five components and **b** its SRG representation

defined in Fig. 1, i.e., $\Sigma_R = \{R_{IC}, R_T, R_{IS}, R_P, R_C\}$ and $2^{\Sigma R}$ is the power set of ΣR. For a candidate object, T_E is defined as $T_E : E \rightarrow \Sigma R$. But for a source object, T_E is defined as $T_E : E \rightarrow 2^{\Sigma R}$, since the relationship between two primitives in the user-drawn object can be ambiguous and can be any combination of the members in ΣR.

(v). $E' \subseteq E$ and $E' = \{(v_1, v_2) | v_1, v_2 \in V, (v_1, v_2) \in E,$ and $v_1 R_P v_2,$ or $v_1 R_T v_2,$ or $v_1 R_C v_2\}$.

(vi). G_E is a function defined as $G_E : E' \rightarrow \Sigma g$, where Σg is the symbolic label set of the four reference directions, which are horizontal, vertical, and two diagonal directions.

(vii). P_E is defined as $P_E : E' \rightarrow \{-1, +1\}$, where $P_E((v_1, v_2)) = +1$ if $v_1 R_{BEF} v_2$ or -1 if $v_2 R_{BEF} v_1$, respectively.

For example, considering the graphic object in Fig. 4, whose components are labeled as $\{A, B, C, D, E\}$, its SRG representation is $G = \{$

$$
\begin{aligned}
V &= \{A, B, C, D, E\},\\
E &= \{(A,B),(A,C),(A,D),(B,C),(B,D),(B,E),\\
&\quad (C,D)\},\\
T_A &= \{(A, T_{Line}),(B, T_{Line}),(C, T_{Line}),(D, T_{Line}),\\
&\quad (E, T_{Arc})\},\\
T_E &= \{((A,B), R_{IC}),((A,C), R_{PA}),((A,D), R_{IC}),\\
&\quad ((B,C), R_{IC}),((B,D), R_{PA}),((B,E), R_T),\\
&\quad ((C,D), R_{IC})\},\\
E' &= \{(A,C),(B,D),(B,E)\},\\
G_E &= \{((A,C), Horizontal),((B,D), Vertical),\\
&\quad ((B,E), Vertical)\},\\
P_E &= \{((A,C),1),((B,D),1),((B,E),1)\}\}.
\end{aligned}
$$

Definition 2. Partial match: Given two SRGs $G_1 = (V_1, E_1, T_{A1}, T_{E1}, E'_1, G_{E1}, P_{E1})$ and $G_2 = (V_2, E_2, T_{A2}, T_{E2}, E'_2, G_{E2}, P_{E2})$, define $m = ||V_1||$, $n = ||V_2||$, and $m \leq n$. Define $V_1 = \{v_{11}, v_{12}, \ldots v_{1m}\}$ and $V_2 = \{v_{21}, v_{22}, \ldots v_{2n}\}$. For a given function

$f : [1 \dots m] \to [1 \dots n]$, satisfying $\forall i, j \in [1 \dots m]$, where $i \neq j$, $f(i) \neq f(j)$, define

(i). $F_V : V_1 \to V_2$ as $F_V(v_{1i}) = v_{2f(i)}$, for any $i \in [1 \dots m]$.
(ii). $F_E : E_1 \to V_2 \times V_2$ as $F_E((v_i, v_j)) = (F_V(v_i), F_V(v_j))$, where $(v_i, v_j) \in E_1$ for any $i, j \in [1 \dots m]$.
G_1 is partially matched with G_2, denoted by $G_1 PM_f G_2$, if the following constraints are met:
Constraint 2.1: $\forall v \in V_1, T_{A1}(v) = T_{A2}(F(v))$.

Constraint 2.2: $\forall e \in E_1, F_E(e) \in E_2$.

Constraint 2.3: $\forall e \in E_1, T_{E1}(e) \supseteq T_{E2}(F_E(e))$.

Constraint 2.4: $\forall e_i, e_j \in E_1'$ where $G_{E1}(e_i) = G_{E1}(e_j), G_{E2}(F_E(e_i)) = G_{E2}(F_E(e_j)), P_{E1}(e_i) \cdot P_{E2}(F_E(e_i)) = P_{E1}(e_j) \cdot P_{E2}(F_E(e_j))$.

Obviously, partial match is an asymmetric relation and depends on the function f. The derived function F_V gives a mapping from the graphic components in the SSO to those in the CRO. The derived function F_E gives a mapping from each spatial adjacency relation in the SSO to that in the CRO. Constraint 2.1 checks whether each graphic primitive type is preserved after mapping. Constraints 2.2 and 2.3 check whether each spatial adjacency relation between graphic primitives in the SSO is "acceptable" to the CRO. If the actual relation in the CRO appears in the SSO, this mapping will be accepted. Constraint 2.4 checks whether the relative position relations inside the same group are preserved after the mapping regardless of the reference direction. For example, in Fig. 5 we find two graphic objects and their SRG representations. Figure 5e demonstrates the partial matching between the two graphic objects.

Now, the problem of whether the SSO is a part of the CRO can be solved in the following steps.

Step 1: Obtain the SRGs of the SSO and the CRO, denoted by G_1 and G_2, respectively.

Step 2: Denote the numbers of nodes in G_1 and G_2 by m and n, respectively (i.e., $m = ||V_1||$ and $n = ||V_2||$),if $m \leq n$, enumerating all possible functions $f : [1 \dots m] \to [1 \dots n]$, satisfying $\forall i, j \in [1 \dots m]$, where $i \neq j$, $f(i) \neq f(j)$.

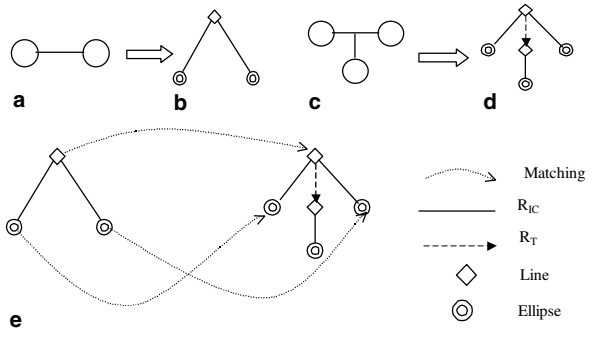


Fig. 5a–e. Example of partial match of SRGs

Step 3: If there exists such a function f satisfying $G_1 PM_f G_2$, we regard the SSO as part of the CRO. Otherwise, we regard the SSO as not part of the CRO.

Obviously, to enumerate all such functions f is equivalent to a partial permutation problem. This classical problem has a computational cost as high as P_n^m. However, since most permutations are illegal mappings, they can be neglected directly during the permutation process according to the four constraints we have defined for partial match. Hence, this partial permutation is a constrained one and referred to as a *constrained partial permutation*. Its computational cost can be controlled under an acceptable size, which mainly depends on various combinations of the SSO and CRO themselves. As shown in our experiments, this strategy is practical for real-time interaction in our prototype system – SmartSketchpad.

Definition 3. Constrained partial permutation: Given two positive integers, m and n, $m \leq n$. Select m integers from $[1 \dots n]$ and then rank them in a list, denoted by $B_1 B_2 \dots B_m$, 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 $(1 \dots n)$ at these positions in the list. Enumerate all possible such lists so that they satisfy one or both of the following two constraints:
Constraint 3.1: i cannot be inserted into place k, that is, $B_k \neq i$.
Constraint 3.2: If i has been inserted into place k, j cannot be inserted into place t, that is, if $B_k = i$, then $B_t \neq j$, where $i, j \in [1 \dots n]; k, t \in [1 \dots m]$.

The first constraint rejects some permutations by matching two vertices between G_1 and G_2, that is, the i-th vertex in G_2 (the i-th component of CRO) cannot match the k-th vertex in G_1 (the k-th component in SSO). We refer to it as the *single denying constraint*, written in a *2-tuple* (i, k). The second constraint rejects some permutations by matching two edges between G_1 and G_2. We refer to it as the *pair denying constraint*, written in a *4-tuple* $(i, k) - (j, t)$.

We transform the partial match problem between SRGs into a constrained partial permutation problem. Let $m = ||V_1||$ and $n = ||V_2||$. We obtain the denying constraints (i.e., find the (i, k) pairs for the first denying constraint and the 4-tuples for the second denying constraint) gradually during the enumerating process. Given a permutation $B_1 B_2 \dots B_m$, we can derive a function $f : [1 \dots m] \to [1 \dots n]$, where $f(i) = B_i$ for any $i \in [1 \dots m]$, and we can obtain the two denying constraints according to the following steps.

Step 1: If f is rejected by Constraint 2.1, there must exist $i, i \in [1 \dots m]$, satisfying $T_{V1}(v_{1i}) \neq T_{V2}(v_{2f(i)}))$. Thus we obtain a single denying constraint (B_i, i).

Step 2: If f is rejected by Constraint 2.2 or 2.3, there must exist $i, j \in [1 \dots m]$, where $(v_{1i}, v_{1j}) \in E_1$, satisfying $(v_{2f(i)}, v_{2f(j)}) \notin E_2$, or there must exist an $R \in \Sigma_R$, so that $R \in R_{E2}((v_{2f(i)}, v_{2f(j)}))$ and $R \notin R_{E1}((v_{1i}, v_{1j})$. Thus we obtain a pair denying constraint $(B_i, i) - (B_j, j)$.

In order to acquire all possible permutations, we regard $B_1B_2 \dots B_m$ as an m-digit n-cardinality integer (i.e., B_k can be a number between 1 and n), in which all digits are different, and then enumerate all such m-digit integers from the smallest, i.e., $123 \dots m$, 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 larger than the current one. The denying constraints will be generated gradually. If the current permutation is $b_1b_2 \dots b_{k-1}ib_{k+1} \dots b_m$, an m-digit n-cardinality integer with no equal digits, and it is rejected by the single denying constraint (i,k), we directly skip all permutations with the form $b_1b_2 \dots b_{k-1}iB_{k+1} \dots B_m$, where B_x $(x = k+1 \dots m)$ belongs to the set of $\{1,2,3,\dots n\} - \{b_1, b_2, \dots b_{k-1}, i\}$ and $B_p \neq B_q$ $(p,q = k+1 \dots m)$, since these will definitely be illegal mappings between G_1 and G_2. If the current permutation is $b_1b_2 \dots b_{k-1}ib_{k+1} \dots b_{t-1}jb_{t+1} \dots b_m$, where $t > k$, and it is rejected by the pair denying constraint $(i,k) - (j,t)$, we directly skip all permutations with the form $b_1b_2 \dots b_{k-1}ib_{k+1} \dots b_{t-1}jB_{t+1} \dots B_m$, where B_x $(x = t+1 \dots m)$ belongs to the set $\{1,2,3,\dots n\} - \{b_1, b_2, \dots b_{k-1}, i, b_{k+1}, \dots, b_{t-1}, j\}$. When this is done, many illegal mappings (permutations) can be pruned directly in the enumerating process.

4.3 Similarity metric

From the above discussion we know that, given an SSO and a CRO represented by their SRGs G_1 and G_2, respectively, each legal mapping/permutation B corresponds to a full match of G_1 in G_2, i.e., full match of the SSO in the CRO. Hence we can calculate the similarity (both visual and structural) between the SSO and the CRO based on the match of their spatial relations and corresponding graphic primitives in the match. Intuitively, the overall similarity of two composite objects is a weighted sum of the similarities of their corresponding primitive objects. For two SRGs, their similarity can be decided by the mapping that can generate the maximum similarity among all possible mappings.

For simplicity, we first define the similarity of two matched primitives P_1 and P_2 according to the relative difference of their length, e.g.,

$$Sim(P_1, P_2) = \frac{\min(L(P_1), L(P_2))}{\max(L(P_1), L(P_2))}, \qquad (1)$$

where $L(P)$ is the length (for line/arc segment) or perimeter (for ellipse) of a graphic primitive P. Note that the similarity value defined in Eq. 1 is normalized (e.g., between 0 and 1).

Next, the similarity between G_1 and G_2 in a given match (legal mapping/permutation B) is denoted by $Sim_B(G_1, G_2)$ and defined as the weighted sum of the similarities of all pairs of matched primitives, as follows.

$$Sim_B(G_1, G_2) = \sum_{1}^{m} w_i Sim(P(v_{1i}), P(v_{2B_i})), \qquad (2)$$

where $P(v_{1i})$ is the primitive at the i-th node in G_1 (denoted by v_{1i}), and $P(v_{2B_i})$ is the primitive at the matching counterpart of v_{1i} in G_2 (denoted by v_{2B_i}), and $w_i$is the weight of $P(v_{1i})$, defined as follows:

$$w_i = \frac{L(P(v_{1i}))}{\sum_{j=1}^{m} L(P(v_{1j}))}, i \in [1 \dots m], v_{1i} \in G_1. \qquad (3)$$

Then, the final similarity between G_1 and G_2 (and between their corresponding SSO and CRO) is defined as the maximal similarity between G_1 and G_2 under all possible mappings/permutations, as follows:

$$Sim(G_1, G_2) = \max_{B \in \psi(G_1, G_2)} Sim_B(G_1, G_2), \qquad (4)$$

where $\psi(G_1, G_2)$ is the set of all legal mappings (permutations) between G_1 and G_2.

The above similarity metric between SSO and CRO is employed to rank the candidate objects that a user intends to draw by sketching a freehand rough object in our experiments presented in the next section, from which we can see that the recognition performance is fairly good.

5 Experimental results

5.1 Experimental environments

In this experiment, we simulate the sketch-based graphics input and recognition process in practical applications. First, 97 composite graphic objects, as shown in Fig. 6, are created to form a database. All these graphic objects have no more than 15 components. Second, we use these objects to generate queries randomly and match these queries with those in the database. For a given graphic object, which has m components, we 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 noise, and a completion rate τ_C, which is used to simulate the incomplete form.

Algorithm 2: Query generating: Given a graphic object with m components and its completion rate τ_C:

Step 1: Randomly select $m * \tau_C$ components.

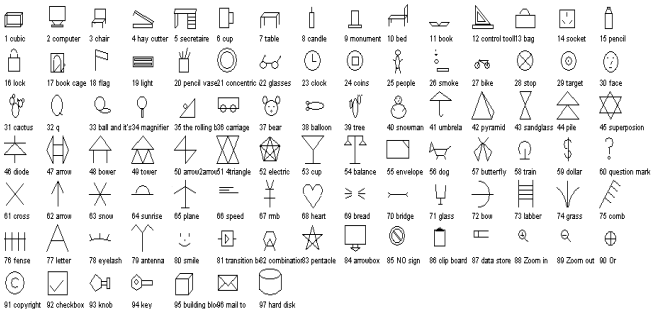
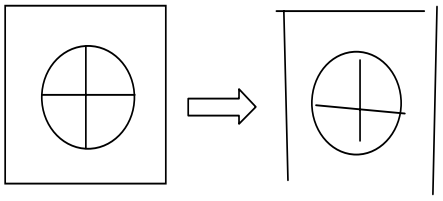


Fig. 6. Composite graphic objects we created for experimentation



a **b**

Fig. 7a,b. An example of generated query object from a regular object

Step 2: Simulate noise for each component.

Step 2.1: Randomize 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: Randomize 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 $m * \tau_C$ components as a whole group, then shift, rotate, and rescale this group, randomly, to form a query q.

Figure 7a shows a regular object stored in the database, and Fig. 7b is the generated query object for this regular object 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 CROs in the database according to their similarities to q_i in descending order. Denote the position of the *i-th* object itself in the ranking list by $Rank_i$, e.g., $Rank_i$ equals 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

$$R_n = \sum_{i=1}^{Total\_Query\_Count} \frac{RANK_{\le n}(q_i)}{Total\_Query\_Count} \quad (5)$$

where (6)

$$RANK_{\le n}(q_i) = \begin{cases} 1 & if\ (RANK_i \le n) \\ 0 & if\ (RANK_i > n) \end{cases} \quad (7)$$

5.2 Performance evaluation

The experimental environment is Pentium III 450 CPU, 256 MB memory, Windows 2000, Visual C++ 6.0. In the experiment, τ_n is set to 0, 0.1, and 0.2, respectively, to simulate different drawing situations. $\tau_n = 0$ is set to simulate a very formal drawing, which has a little noise. $\tau_n = 0.1$ is to simulate an ordinary user-sketched drawing, which has some noise. $\tau_n = 0.2$ is to simulate a very sketchy drawing, which has much more noise when the user draws freely. On the other hand, τ_C increasess from 0.5 to 1.0, simulating different completion status. R_i is defined in Eq. 5. R_i as a percentage of different τ_n and τ_C is shown in Fig. 8.

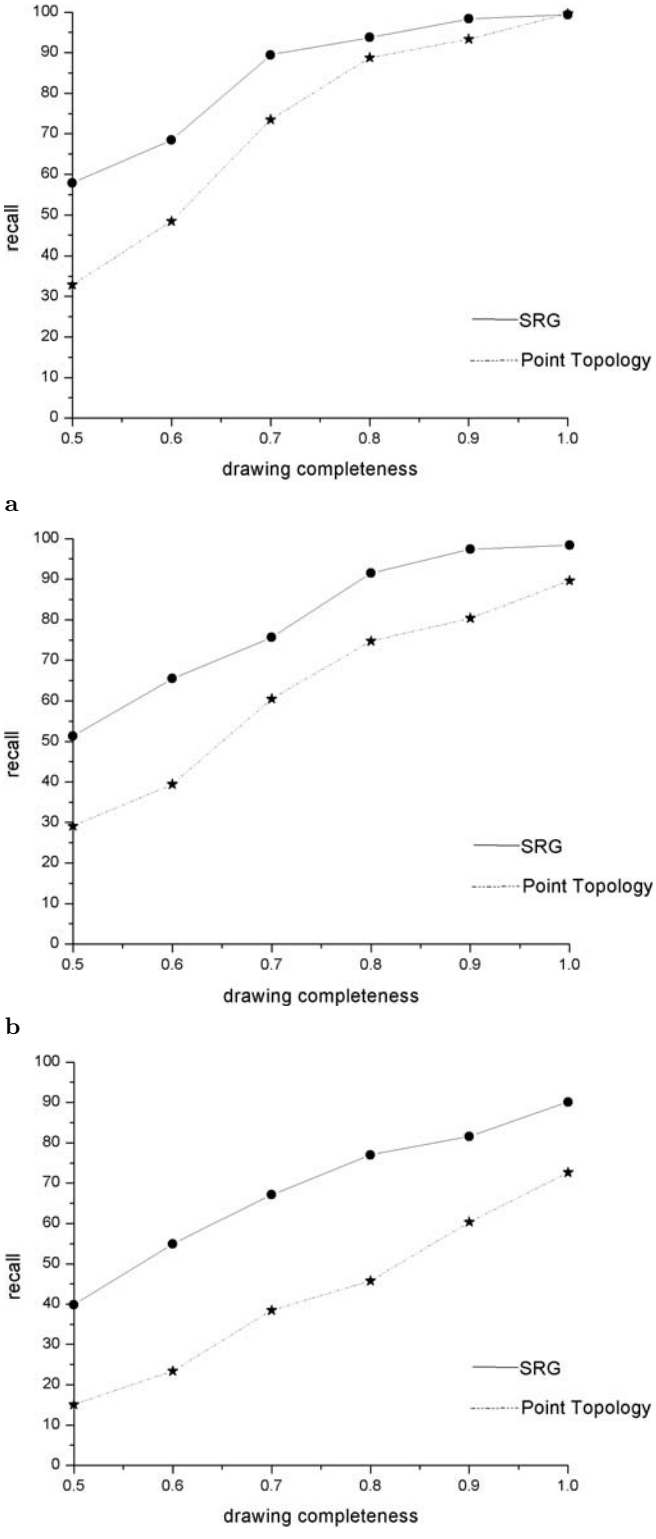


Fig. 8a–c. Performance comparison between SRG-based approach and point-topology- based approach

From Fig. 8 we see that SRGs are relatively sensitive to noise when there are relatively few graph nodes

Table 1. Time (in milliseconds) used in the graphics recognition process (m is the number of query object components and n the number of database object components)

m \ n	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
2	0.41	0	0.20	0.19	0.08	0.08	0.22	0.11	0.14	0.06	0	0.36	0
3		0.03	0.07	0.15	0.27	0.11	0.09	0.09	0.39	0.29	0.59	0	0.29
4			0.15	0.12	0.09	0.12	0.20	0.14	0.41	0.35	0.28	0.28	0.83
5				0.12	0.11	0.25	0.27	0.41	0.70	1.07	0.54	0.77	3.08
6					0.16	0.16	0.30	0.50	0.88	1.01	0.36	4.02	5.90
7						0.14	0.25	0.73	0.72	1.08	0.21	2.13	5.86
8							0.33	1.23	2.43	3.51	3.33	54.62	26.08
9								0.56	1.76	1.54	0	4.8	106.32
10									3.11	1.34	0.67	0.33	1
11										1.67	40	0	1.25
12											10	0	0
13												20	80.25
14													7.5

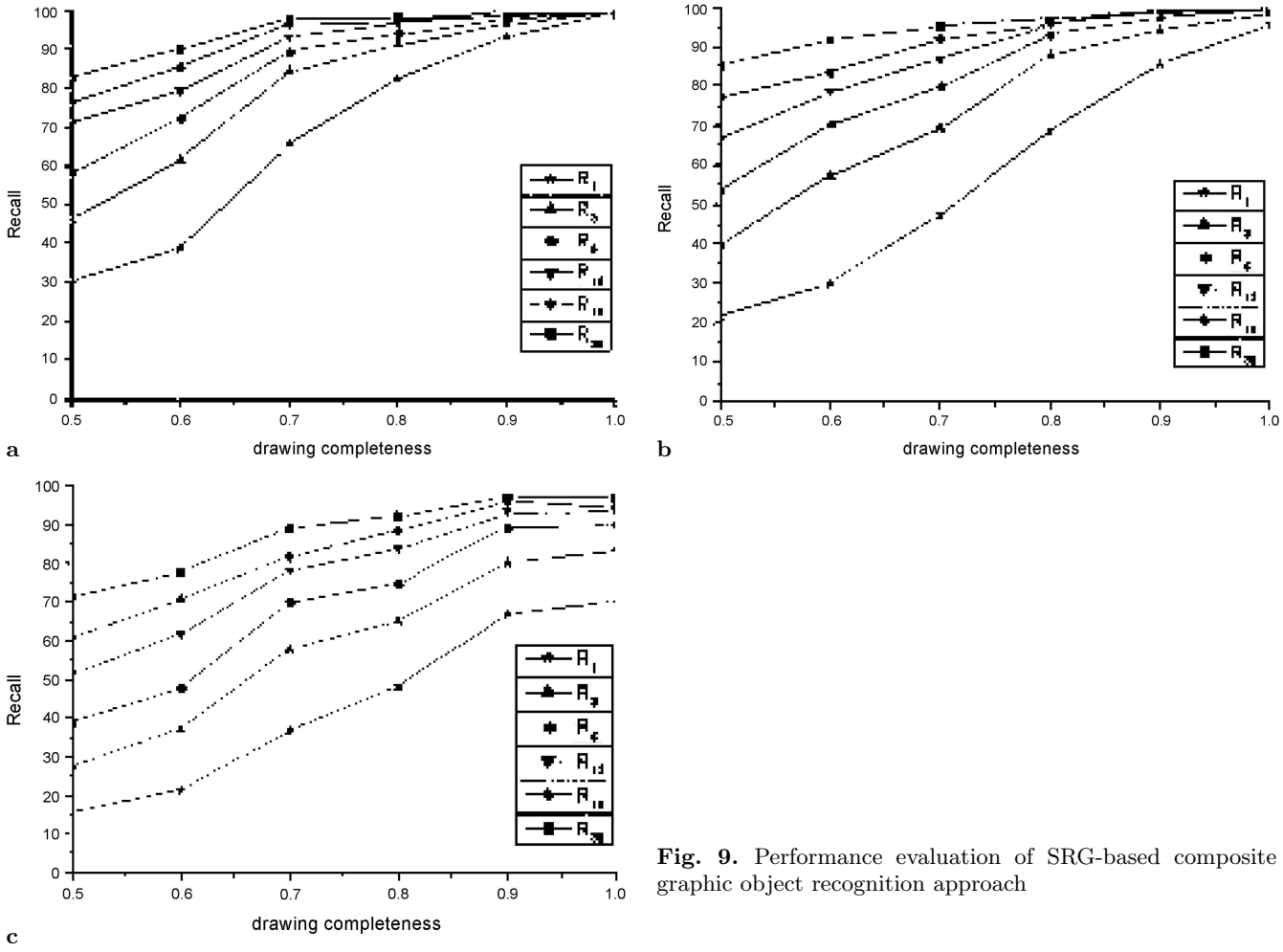


Fig. 9. Performance evaluation of SRG-based composite graphic object recognition approach

(i.e., 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 also see that, when noise is 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 noise for incomplete sketchy input.

The time cost used in the recognition process is also a factor of great concern for evaluating our approach's performance. Especially in our real-time interactive environment, the time cost should be as small as possible.

The time cost in milliseconds of our approach is listed in Table 1.

In Table 1, the blank cells mean that we do not calculate them since the value of m (number of components in the query object) is bigger than that of n (number of components in the object in database), and this conflicts with the definition of partial matching in Definition 2. The zero values in Table 1 are very small real numbers. From Table 1 we find that the maximum time used is 106.32 ms and the minimum time used is far less than 1 ms. Generally, the time cost is much smaller than one tenth of a second. The performance is sufficiently fast for real-time interaction.

We have also compared the performance of the approach proposed in this paper with the previous algorithm used in our prototype system [35], which mainly used the dominant feature points' distribution (referred to as point topology) information [21]. The comparison results are illustrated in Fig. 9. From Fig. 9 we find that the former approach is very sensitive to noise and cannot support the partial matching as well as the SRG-based approach. When there is no noise and the input is completed, the average recalls of the two approaches are almost equal. But on average, SRG has better performance and has some immune ability to noise to some extent.

6 Conclusions

In this paper, we have proposed a spatial relation graph (SRG) for composite graphic object representation and developed an SRG-based approach to composite graphic object recognition. Specifically, we have developed a constrained partial permutation strategy to prune illegal cases and hence reduce the computational cost of matching two SRGs. From the experiment results, we can see that our strategy is practical in real-time sketch-based graphics input and recognition systems.

As a structural (or topological) representation of composite graphic objects, SRG has been proved in our experiments to be effective for precisely matching composite graphic objects, even when there is moderate noise in the input object and when the input object is not completely drawn. SRGs are relatively sensitive to noise when few components have been drawn. As the object becomes more complete, higher recall rates can be achieved. This means that the SRG-based approach can predict a user's intended object from an incomplete input when the noise is relatively small. Compared with our previous approach, which was based on the point topology of graphic objects, the SRG-based approach is more immune to noise and has a better performance when the input is not complete. Hence we conclude that the SRG-based approach is both efficient and effective for online graphics recognition systems, e.g., our sketch-based graphics input system [35].

However, the spatial relations we enumerated in this paper may not be enough to represent all spatial relations in all types of graphic objects. In some very rare cases, the matching time is still very high for real-time interaction. Therefore, in our future work, more spatial relations should be investigated and the constrained partial permutation strategy should be improved further to prune more illegal mappings earlier.

Acknowledgements. We are sincerely grateful to Professors Andrew Yao and Frances Yao for their advice on this paper. We also thank Professor Pan Zhigen and Miss Hu Weixi for their kind help. The work described in this paper was partially supported by a grant from the National Natural Science Foundation of China (Project No. **69903006**, **60373065**) and by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 1073/02E).

References

1. Almohamad HA, Duffuaa SO (1993) A linear programming approach for the weighted graph matching problem. IEEE Trans PAMI 15(5):522–525
2. Alvarado C, Oltmans M, Davis R (2002) A framework for multi-domain sketch recognition. In: Proceedings of the AAAI spring symposium on sketch understanding, pp 1–8
3. Chang S-K, Shi QY, Yan CW (1987) Iconic indexing by 2-D strings. IEEE Trans PAMI 9(3):413–428
4. Christmas WJ, Kittler J, Petrou M (1995) Structural matching in computer vision using probabilistic relaxation. IEEE Trans PAMI 17(8):749–764
5. Davis R (2002) Sketch understanding in design: overview of work at the MIT AI Lab. In: Proceedings of the 2002 AAAI spring symposium on sketch understanding, pp 24–31
6. Foggia P, Sansone C, Vento M (2001) A database of graphs for isomorphism and sub-graph isomorphism benchmarking. In: Proceedings of the 3rd IAPR-TC15 workshop on graph-based representations in pattern recognition, Ischia
7. Fonseca MJ, Jorge JA (2000) Using fuzzy logic to recognize geometric shapes interactively. In: Proceedings of the 9th IEEE conference on fuzzy systems, 1:291–196
8. Fonseca MJ, Pimetal C, Jorge J (2002) CALI: an online scribble recognizer for calligraphic interfaces. In: Proceedings of the 2002 AAAI spring symposium on sketch understanding, pp 51–58
9. Gold S, Rangarajan A (1996) A graduated assignment algorithm for graph matching. IEEE Trans PAMI 18(4):522–525
10. Gross MD (1996) The electronic cocktail napkin – a computational environment for working with design diagrams. Des Stud 17:53–69
11. Gross MD, Do EY (1996) Ambiguous intentions: a paper-like interface for creative designing. In: In: Proceedings of the ACM symposium on user interface software and technology, pp:183–192
12. Hammond T, Davis R (2002) Tahuti: a geometrical sketch recognition for UML class diagrams. In: Proceedings of the 2002 AAAI spring symposium on sketch understanding, pp 59–66
13. Jin XY, Liu WY, Sun JY, Sun ZX (2002) Online graphics recognition. In: Proceedings of the Pacific Graphics conference, Beijing, October 2002, pp 256–265

14. Kim IJ, Kim JH, Liu CL (1999) Stroke-guided pixel matching for handwritten Chinese character recognition. In: Proceedings of the 5th international conference on document analysis and recognition
15. Landay J, Myers B (1995) Interactive sketching for the early stages of user interface. In: Proceedings of the ACM CHI 95 conference on human factors in computing systems, pp 43–50
16. Landay J, Myers B (2001) Sketching interfaces: toward more human interface design. IEEE Comput 34(3):56–64
17. Lee SY, Hsu FJ (1990) 2D C-string: a new spatial knowledge representation for image database systems. Patt Recog 23(10):1077–1087
18. Lee SW, Kim YJ (1995) A new type of recurrent neural network for handwritten character recognition. In: Proceedings of the 3rd international conference on document analysis and recognition, 1:38–42
19. Li C, Yang B, Xie W (2000) On-line hand-sketched graphics recognition based on attributed relation graph matching. In: Proceedings of the 3rd world congress on intelligent control and automation, Hefei, China, pp 2549–2553
20. Liu WY (2004) On-line graphics recognition: state of the art. Lecture notes in computer science. Springer, Berlin Heidelberg New York. (Selected and revised papers from Proceedings of GREC2003, July 2004, Spain (in press)
21. Liu WY, Qian W, Xiao R, Jin XY (2001) SmartSketchpad – an online graphics recognition system. In: Proceedings of the ICDAR2001, Seattle, pp 1050–1054
22. Liu WY, Jin XY, Sun ZX (2002) Sketch-based user interface for inputting graphic objects on small screen devices. Lecture notes in computer science, vol 2390. Springer, Berlin Heidelberg New York. (Selected and revised papers from Proceedings of GREC2001, 7–8 September 2001, Kingston, Canada)
23. Lladös J, Martí E, José J (2001) Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. IEEE Trans PAMI 23(10)::1137–1143
24. Mehlhorn K (1984) Graph algorithm and NP-completeness. Springer, Berlin Heidelberg New York
25. Messmer BT (1995) Efficient graph matching algorithms. PhD thesis, University of Bern, Switzerland
26. Messmer BT, Bunke H (1998) A new algorithm for error-tolerant sub-graph isomorphism detection. IEEE Trans PAMI 20(5):493–504
27. Müller S, Eickeler S, Rigoll G (1998) Image database retrieval of rotated objects by user sketch. In: Proceedings of the IEEE workshop on content-based access to image and video libraries (CBAIVL), Santa Barbara, CA pp 40–44
28. Osuna E, Freund R, Girosi F (1997) Training support vector machines: an application to face detection. In: Proceedings of CVPR 97
29. Petrakis EGM (2002) Design and evaluation of spatial similarity approach for image retrieval. Image Vision Comput 20:59–76
30. Plamondon R, Guerfali W, Lalonde M (1999) Automatic signature verification: a report on a large-scale public experiment. In: Proceedings of the 9th biennial conference, Singapore, pp 9–13
31. Pontil M, Verri A (1998) Support vector machines for 3D object recognition. IEEE Trans PAMI 20(6):637–646
32. Rabiner LR (1989) Tutorial on hidden Markov model and selected applications in speech recognition. Proc IEEE 77(2):257–285
33. Ranganath HS, Chipman LJ (1992) Fuzzy relaxation approach for inexact scene matching. Image Vision Comput 10(9):631–640
34. Saund E (2003) Finding perceptually closed paths in sketches and drawings. IEEE Trans Patt Anal Mach Intell 25(4):475–491
35. Sun ZX, Xu XG, Sun JY, Jin XY (2003) Sketch-based graphic input tool for conceptual design. J Comput Aided Des Comput Graph (in Chinese) 15(9):205–206
36. Xu XG, Liu WY, Jin XY, Sun ZX (2002) Sketch-based user interface for creative tasks. In: Proceedings of the 5th Asia Pacific conference on computer human interaction, Beijing, pp 560–570

Xu Xiaogang received his B.Sc. and M.Engg. degrees in computer science from the Nanjing University, Nanjing, China, in 2000 and 2003, respectively. Prior to this he worked as a research assistant at Nanjing University for 3 years and at the City University of Hong Kong for 3 months. Currently he is a software engineer in the Motorola Global Software Group. His research interests include graphics recognition, pattern recognition, sketch understanding, and multimodal user interface.

Sun Zhengxing received his B.Engg. from Southeast University, Nanjing, China, in 1985 and his M.Engg. and Ph.D. from Nanjing University of Aeronautics and Astronautics, Nanjing in 1992 and 1996, respectively. He worked at the Computer Center of Xuzhou Constructional Machinery Research Institute as a software engineer from 1985 to 1993, and was a postdoc researcher in the Department of Computer Science, Nanjing University, from 1996 to 1999. He is now full professor in the Department of Computer Science, Nanjing University. His research interests include multimedia information retrieval, sketching intelligence systems, Web services engineering, data warehouse and data mining, and enterprise computing. He has authored or coauthored over 80 publications in these areas.

Peng Binbin received his B.Sc. in mathematics from Nanjing University, Nanjing, China in 2000 and his M.Engg. in computer science from the same university in 2003. Prior to this he worked as research assistant at Nanjing University for 3 years. Currently he is a research assistant at the City University of Hong Kong. His research interests include machine learning, graphics recognition, pattern recognition, sketch understanding, and user adaptation.

Jin Xiangyu received his B.Sc. and M.Engg. in computer science from Nanjing University, Nanjing, China, in 1999 and 2002, respectively. Prior to this he worked as a visiting student at Microsoft Research Asian in 2001. Currently he is a Ph.D. student at the University of Virginia. His research interests include multimedia information retrieval, user interface, Web publication, and digital library.

Liu Wenyin received his B.Engg. and M.Engg. in computer science from the Tsinghua University, Beijing in 1988 and 1992, respectively, and his D.Sc. from the Technion, Israel Institute of Technology in 1998. Prior to this he worked for 3 years at Tsinghua as a faculty member and at Microsoft Research China/Asia as a full-time researcher for another 3 years. He is now assistant professor in the Department of Computer Science at the City University of Hong Kong. Liu Wenyin played a major role in developing the MDUS system, which won first place in the Dashed Line Recognition Contest in 1995. In 1997, he won third prize in the ACM/IBM First International Java Programming Contest (ACM Quest for Java97). In 2003, he was awarded the ICDAR Outstanding Young Researcher Award by IAPR for his significant contribution to the area of graphics recognition, engineering drawing recognition, and performance evaluation. He is a senior member of IEEE.