Shape-Based Visual Query Rewriting

Georges Chalhoub, Richard Chbeir, Kokou Yetongnon
Computer Science Department, LE2I – Bourgogne University
BP 47870 21078 Dijon - France
gchalhoub@bdl.gov.lb
{richard.chbeir, Kokou.yetongnon}@u-bourgogne.fr

ABSTRACT

A visual query is based on pictorial representation of conceptual entities and operations. One of the most important features used in visual queries is the shape. Despite its intuitive writing, a shape-based visual query usually suffers of a complexity processing related to two major parameters: 1-the imprecise user request, 2-shapes may undergo several types of transformation. Several methods are provided in the literature to assist the user during query writing. In this paper, we present a new cooperative approach based on the shape neighborhood concept allowing the user to rewrite a shape-based visual query according to his preferences with high flexibility in terms of including (or excluding) only some shape transformations and of result sorting.

1. INTRODUCTION

A visual language is based on pictorial representation of conceptual entities and operations through which users compose iconic or visual sentences. Several visual features (such as icons, predefined shapes, primitive shapes, sample images, etc.) can be combined together using spatial, temporal and logical operators. Shape-based queries are widely used in visual languages due to their simplicity and intuitivity. Three main categories of shape-based visual languages are provided in the literature: Iconic-based [1], Sketch-based [10], and Query By Image [8]. Using these user-friendly languages, the user can easily visualize and graphically query the database. However, several limitations are identified and related to the use of each one of these methods. For instance, when using iconic-based languages, the query may encounter some ambiguities when the operators and objects number increases. Query by image queries are very restrictive when the user does not have a sample image expressing his needs. To handle these limitations and make the retrieval process more cooperative, several techniques have been provided in the literature [5].

Widely used in several search engines and for textual data, the query rewriting technique has been studied in several domains [3]. The relevance feedback is one of the query rewriting techniques [9]. It aims at providing users the opportunity to evaluate search results by selecting relevant (or irrelevant) ones. The system can then iteratively rewrite the initial query in function of the selected sets given by the user after each step. However, most of current approaches do not allow the user to specify neither the degree of relevance (or irrelevance) of each result, nor the order of searching and/or displaying retrieval results. In essence, shape retrieval is a complex task due to several transformations (occlusion, articulation, rotation, translation, scaling, etc.) that a shape may undergo. When retrieving similar shapes, current techniques are able to consider only a set of domain-related transformations within a predefined execution order. Moreover, in order to keep the retrieval interfaces user friendly, they attempt, even when using relevance feedback techniques, to simplify the user intervention by limiting the input or feedback parameters which is very restrictive when formulating complex queries (which transformations to include or to exclude?, which sorting order?, etc.). In [3], an interesting rewriting approach has been provided for multimedia queries. The authors have defined a relaxation and a constraint functions to rewrite only textual-oriented queries using the user profile. In this paper, we extend their approach to shape features and define a formal language for shape rewriting. Here, the relaxation function allows considering all types of shape transformation (stretching, occlusion, rotation etc.), while the constraint function aims at:

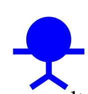
Including and/or excluding shapes from the relaxation result,

Assigning an order to relaxation results according to the user requests.

2. MOTIVATION

To explain the motivation of this work, let us consider the following example: A journalist takes using a digital camera some snapshots in front of the finish line of the 100, 200, and 400 meters men competitions in the 10th IAAF World Championships in Athletics. Afterwards, he stores the captured pictures in an image database (or repository) without any annotation. The journalist uses a retrieval tool that extracts from the stored images a set of corresponding shape representations as shown in figure 1. The tool provides a shape-based sketch and iconic-based image retrieval interface, with a relevance feedback technique to refine the user query. It uses global similarity measure between shapes (figure 2) allowing the user to express the similarity degree by giving a similarity threshold[1] $\varepsilon \in [0, 1]$.

To write his weekly report, the journalist wants to look for only Golden winners' shots taking at the final stage of the competition. He formulates his query Q by drawing a sample shape (imagining a typical one when wining a competition at the arrival stage) as follows:

Q: 

The query results expected by the journalist must contain the following shapes:

- *Shape D which is the initial query*,
- *Shape I and J representing an athlete raising two hands*,
- *Shape B representing an athlete raising only one hand*

[1] is related to the number of links to consider when computing the similarity.

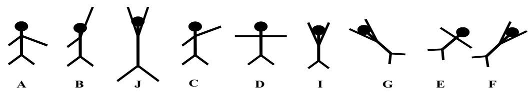


Fig. 1. A set of shapes representation in the database.

In the following, we give a traditional technique scenario describing the steps followed by the journalist attempting to obtain the expected relevant results:

1. The journalist gives the initial query Q (shape D) using a threshold ε_1 (we assume here that the distance within ε_1, gives one neighborhood link when computing similar shapes).
2. The system formulates the query and returns the most similar or closest shapes (D, C, E and I) as appearing in figure 2
3. The journalist marks E and I as relevant shapes, and C as irrelevant one
4. The system rewrites the query by excluding similar shapes to C, and including similar ones to E and I. The new result contains
5. shape D and J (which are expected)
6. shape F and G (close to I) which are unexpected
7. The journalist may mark new irrelevant and relevant shapes until having shapes D, I and J.

The result may never contain shape B (eliminated when the journalist has eliminated C) and the silence rate, if best, would be of 1/(1+3)=0.25.

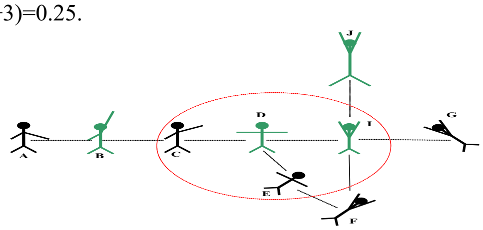


Fig. 2. Similarity links between shapes.

In the above scenario, we attempted to show how most approaches using relevance feedback would usually work when cooperating with the end-user to retrieve expected results. However, we do believe that a shape-based visual query approach should be more flexible and provide:

- *Higher expressive similarity measure:* A shape may undergo several transformations. The user should be able to exclude all shapes resulting from one or more transformations. In our example, the journalist would have excluded the rotation transformation results and thus reduced the feedback interaction numbers
- *Customized inclusion and exclusion parameters*: The user may want to exclude from the result a shape without excluding its neighborhoods. In our example, the journalist would have excluded the shape C without excluding D and B, which are neighborhood shapes of C.
- *Customizable retrieval result:* In our example, the journalist is more interested in the shape D and I than the shape B.

3. RELATED WORK

Widely used in several Information Retrieval Systems (IRS), query rewriting (or reformulation) techniques allow a system to cooperate with the user during the retrieval phase in order to better meet his requirements. Two main rewriting techniques categories are identified: *Query-oriented techniques* [8] and *User-oriented techniques* [6]

Iconic languages allow the user to formulate queries using predefined icons representing domain-related objects and operators. In [1], CIGALES language allows the user to formulate queries using predefined icons.

Using sketch languages, the user can formulate his query without the constraints of predefined icons. In [10], using the Sketch! Language, the user formulates his query by drawing spatial objects and operators..

Query By Image (QBI) technique allows the user to provide a set of query images (usually one image) similar to those stored in the corpus. It has been studied and integrated in several retrieval systems and DBMS. In [8], the authors describe a visual query language (VQL) among time series data. In the literature, other studies aim at incorporating human perception subjectivity into the retrieval process and providing users the opportunity to evaluate retrieval results. In [9], the authors define a relevance feedback method that takes as input a query image and a list of images that have been marked as either relevant or irrelevant by the user.

It goes without saying that all the above works are interesting and facilitate the query formulation using a visual interface and content-based retrieval with the possibility of the user relevance feedback. However, in relevance feedback techniques, the user can qualify the object as relevant or irrelevant. However, he has no choice to specify the degree of relevance or irrelevance. In addition, during the retrieval process, shape matching is expressed according to shape similarity without considering each shape transformation. In some situations, this would increase the retrieval steps and silence rate.

4. SHAPE BASED QUERY REWRITING

In this paper, we extend the textual-oriented rewriting approach presented in [3] by considering the shape feature and providing a flexible formal language for shape-based query rewriting. An initial user query Q is formally rewritten into Q^A as follows:

$$Rewriting(Q, \{R, T\}, \{C\}) \rightarrow Q^A$$

Where R is a shape transformation, T is a threshold, and C is a constraint set.

Our proposal is independent of the methods and algorithms used to represent or retrieve a shape. However, two main properties in the algorithms are required to rewrite the shape query using our approach:

- Uniqueness: the algorithm must associate to each shape only one representation (a graph, tree, etc.).
- Cost calculation: the algorithm must be able to calculate the cost of matching between two shapes representations.

In the following, we give a definition concerning the dissimilarity cost matching between two shapes[2]. After, we define the concept of shape neighborhood. The cost and the neighborhoods are defined with respect to each shape transformation. To classify neighborhood elements, we affect to each one a weight expressing the closeness to the original shape. Based on these definitions, we explain our rewriting approach and study several current shape representation methods provided in the literature.

[2] The given cost definition is based on graph matching algorithms.

The first paragraph in each section should not be indented, but all following paragraphs within the section should be indented as these paragraphs demonstrate.

4.1 Definitions

Definition 1: Matching cost

- A matching cost is calculated to measure the dissimilarity between one shape *A* and another shape *B* having undergone a transformation *R*. To represent a shape, two approaches are provided in the literature: curve-based [2] and graph-based approaches [7]. For instance, in graph-based approach [7], each shape is represented as a graph and the dissimilarity between two shapes is measured by calculating the matching cost of their corresponding graphs. The matching cost between two shape graphs A and B can be computed as with respect to *Contour* and *transformation matching*. Graph matching operations can be done using the A*LIKE algorithm [4].

Formally, we note the cost of matching between two shapes A and B, $\mathrm{Cost}_{Mt}(A,B)$ considering the contour matching cost_c and the transformation R, cost_R as follows:

$\mathrm{Cost}_{Mt}(A,B) = Cost_C(A,B) + Cost_R(A,B)$

Definition 2: Shape Neighbourhood

The neighborhood of the shape A according to a transformation R and a threshold ε_R, is defined by the set $V_R(A, \varepsilon_R)$ as follows:

- The matching cost between an element (shape) B belonging to $V_R(A, \varepsilon_R)$ and the shape A is less than or equal to ε_R. In other words, B is considered as a transformation of the original shape A by the transformation R within a threshold ε_R.
- ε_R is a threshold defined with respect to the transformation R. We note that the domain of ε is related to the shape transformation. For instance, if the transformation R is a *shape rotation* then the domain of ε is the interval [0, 360]. For other transformations, the domain of ε is different. Thus, the value of ε should be normalized. This normalization can be done by a linear function for a given ε:

$$\varepsilon_R = f(\varepsilon) = \frac{\varepsilon - \min}{\max - \min + 1}$$

Where max and min are respectively the maximum and the minimum values of ε in each domain.

- The same function is applied to cost normalization.

We represent a formal definition of the neighborhood of a shape A as follows:

$$V_R(A,\varepsilon) = \{B \,/\, B \in R(A) \text{ and } \mathrm{Cost}_{Mt}(A,B) \le \varepsilon_R\}$$

Definition 3: Shape weighting

In the neighbourhood of a shape A, the elements have different *weights.* The weight of an element is defined according to its closeness to the original shape A. The elements in the neighbourhood are sorted according to their weights. To express the weight, we associate to each shape B a positive real value less than or equal to 1. The weight W_B is associated with a shape B as follows:

$$W_B = 1 - \mathrm{Cost}_{Mt}(A,B)$$

You can observe, the weight of a shape B is less than the weight of the original shape A. This weight is useful in shape relaxation to classify the query result according to the closeness to the original shape.

4.2 Rewriting Process

Now, let us explain how we rewrite a shape–based visual query using our approach. The rewriting process based on two principal functions: Relax function F_R and Constraint function F_C. These two functions were defined to relax terms and relations in [3]. We extend their use as follows: The function F_R allows returning a set of relaxed shapes, and F_C controls the returned result of F_R. Shape rewriting can be formally defined as:

Rewrite(shape element) = Rewrite(δ) = $F_C(F_R(\delta))$ = $F_C(\delta')$ = δ'_c

Definition 4: Shape element δ

The shape element **δ** is a triplet (**A, R, T**) where:

- $\mathcal{A}$: is the original shape that the user wants to relax
- $\mathcal{R}$: is the transformation function applied to $\mathcal{A}$
- $\mathcal{T}$: is the relaxation threshold of A in R. T ∈ [0, 1] and represents the maximum distance of a shape B ∈ VR (A) to consider in the relaxation

Definition 5: Relax function FR

We define $\mathbf{F_R}$ as the relaxation function to be applied on an element **δ.** It returns a sorted set **δ'** of couples (shape, weight) related to **A** in descending order. Each couple of **δ'** is a node (value and weight) selected from the neighbourhood of **A** with respect to the transformation **R**. The distance (weight difference) between **A** and a selected shape is less than or equal to **T**. $\mathbf{F_R}$ is formally formulated as follows:

$$F_R: \delta = (\mathcal{A},\mathcal{R},\mathcal{T}) \rightarrow F_R(\delta) = \delta' = \{(\text{shape}, W), \le_W\}$$

Definition 6: Constraint function F_C

Sometimes, the user desires to exclude (or include) some shapes from the result set **δ'.** To accomplish this, we define a constraint set **C** and a constraint function $\mathbf{F_C}$ as follows:

- **C** is a set of shapes. It is represented by a set of couples (δ'_P, W) where δ'_P is a subset of δ'. It contains both the set of shapes to be excluded (or included) from the result, and shapes whose weights are to be modified. δ'_p may contain one or several shapes.
- $\mathbf{F_C}$ is a function that applies C constraints to the result **δ'** as follows:

$$F_C: (\delta', C) \rightarrow F_C(\delta', C) = \delta'_c =$$
$$\delta' - \{(\delta_p', W') \text{ where } (\delta_p', W) \in C \text{ and } 0 \le W < 1\}$$
$$\cup$$
$$\{(\delta_p', W) \text{ where } (\delta_p', W) \in C \text{ and } 0 < W \le 1\}$$

In other words:

- If W = 1 then $\mathbf{F_C}$ includes the shapes of δ_p' to the relaxation result **δ'**,
- If W = 0 then $\mathbf{F_C}$ excludes the shapes of δ_p' from relaxation result **δ'**,
- If 0 < W < 1 then $\mathbf{F_C}$ modifies the weight of the shapes of δ_p' in the relaxation result (if Val exists in the result).

4.3 Discussion

In this section, we show how the query (re)writing of our motivation section example can be done using our approach. Consider now transformations (Occlusion, Rotation, and Stretching) when computing the similarity between shapes (figure 3) allowing the user to give a threshold ε ∈ [0,1] for each transformation measure. Our approach is also applicable if only one similarity measure is used. To obtain expected results, the following steps are applied:

1. The journalist formulates the query Q and gives the following parameters:
   - exclude rotated shapes (ε_R=0)
   - include occluded shapes using ε_O

- include stretched shapes using ε_S

2. The system formulates the query and returns the most similar or closest shapes (D, C, and I)
3. The journalist marks C as irrelevant (without excluding its neighborhood shapes), E and I as relevant result
4. The retuned most similar shapes: D, I, J, B

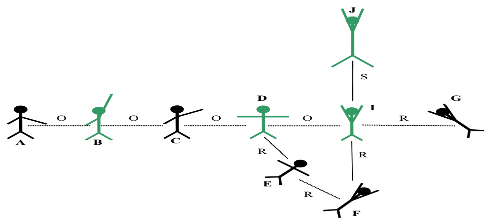


Fig. 3. Transformation links between shapes.

The above scenario shows that our proposal is able to:

- Allow the user to specify results priorities. For instance, if the journalist is interested the most in shape D for all competitors then he associates to the stretching transformation the first priority. The highest priority results are shapes taller or shorter than D within a stretching threshold given by the user.
- Allow the user to exclude a shape, from the result, without excluding its neighborhood shapes. For instance, in step 3 of the above scenario, the journalist excludes shape C without excluding shapes B and D.
- Decrease the silence rate because of its higher expressive power.
- Decrease the user interventions during the search process.

5. IMPLEMENTATION

To validate our approach, we implemented a Java-based prototype[3] able to provide to the user a shape-based retrieval interface dealing with several shape transformations such as *stretching, occlusion, articulation,* and *rotation.* As we mentioned before, curve-based approaches [2] and graph-based approaches [7] are provided in the literature to represent the shape. In our prototype, we adopted a graph-based method called ***shape-axis*** [7]. It consists of representing each shape by a unique axis tree for similarity computation. The shape-axis method can illustrate the skeleton of both the open and the closed curves as mentioned in [7]. In addition, the *shape-axis* representation is sensitive to stretching; and other transformations like occlusion.

For instance, based on matching cost, the shape neighborhood is defined according to the *stretching* transformation neighborhood as follows:

The similarity between A and B is identified by computing the correspondence (commonly called Edge-to-path correspondence) between them using a merging operation. In [7], the authors tried to find the total cost of correspondent segments CostS, and suggested to use a penalty cost for the merging operation. This cost is called CostM that computes the cost of the merged segment and its correspondent node. To compute the similarity between two shapes, the authors proposed to calculate the total cost representing the similarity cost of the compared parts of the two shapes.

The total cost of matching between A and B is represented as follows:

$$Cost_{Mt}(A, B) = Cost_S(A,B) + Cost_M(A, B)$$

[3] Not presented in this paper due to the space limitation

In this way, we are able to define the *Stretching neighborhood* VS of a shape A, using the set of shapes B where the matching cost CostS(A,B) and the merging cost CostM(A, B) are less than the value Cs given by the user. The neighborhood is formally defined as follows:

$$V_S(A, C_s) = \{B \,/\, (Cost_S(A,B) + Cost_M(A,B) \leq C_s)$$

V_S represents the neighborhood set that enables 2 parameters:

1. The original image A to be compared with the articulated image B
2. **C** is the threshold of the exact matching plus the cost the merging operation. C is defined by the user

6. CONCLUSION

In this paper, we proposed a new visual shape-based query rewriting approach. It allows the user to have higher expressive power than traditional shape-based retrieval approaches. Customized inclusion and exclusion parameters are provided to the user when (re)formulating the query. In addition, the retrieval result shapes can be sorted according to the user preferences.

We are currently studying curve-based approaches provided in the literature and how we can integrate them into the prototype. We are also experimenting our prototype using a SVG database with about 600 documents. Our future work will address the integration of physical features like colour and texture into our rewriting approach

7. REFERENCES

[1] Aufaure, P., A High-Level Interface Language for GIS, Journal of Visual Languages and Computing, 6(2) pp: 167-182, 1995.

[2]Boaz J. Super, Improving Object Recognition Accuracy and Speed through Non-Uniform Sampling, Proc. of the SPIE Conference on Intelligent Robots and Computer SPIE Vol. 5267, pp. 228-239, 2003

[3] Chalhoub G., Saad S., Chbeir R. and Yetongnon K., Flexible Multimedia Query Rewriting , Workshop on Distributed Multimedia Databases and Multimedia Adaptation (WDMDMA) in EUROMEDIA'2005, 11 April 2005.

[4] D. Shasha, J. Wang and K. Zhang. Exact and Approximate Algorithm for Unordered Tree Matching. IEEE Trans. Systems,Man, and Cybernetics, 24(4), pp. 668-678, 1994.

[5] Gaasterland, T., Godfrey, P. and Minker, J. An overview of cooperative answering. Journal of Intelligent Information Systems, pp 123-157, 1992.

[6] Google, http://www.google.com (Last visited 11/09/2005)

[7] Liu, T., Geiger, D. Approximate Tree Matching and Shape Similarity. IEEE 17th International Conference on Computer Vision, pp 456-462, 1999.

[8]Long, H. and Leow, W. K. Perceptual consistency improves image retrieval performance , ACM SIGIR, pp. 434-435, 2001

[9] MacArthur, S. D., Brodley, C. E. and Shyu, C. Relevance Feedback Decision Trees in Content-Based Image Retrieval, in Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, pp 68-72, June 2000.

[10] Meyer, B., Beyond Icons: Towards New Metaphors for Visual Query Languages for Spatial Information Systems, Proceedings of the first International Workshop on Interfaces to Database Systems, pp 113-135, 1992.