Shape-Based Visual Query Rewriting

Georges Chalhoub[1], Richard Chbeir[2], Kokou Yetongnon[2]

Computer Science Department, LE2I – Bourgogne University
BP 47870 21078 Dijon - France
[1]gchalhoub@bdl.gov.lb
[2]{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 due to the user certainty or to the vagueness of his need, 2-shapes may undergo several types of transformation like occlusion, rotation, etc. that need to be considered. This is why several methods are provided in the literature to assist the user during query writing. On one hand, relevance feedback technique is widely used to rewrite the initial user query. On the other hand, shape transformations are considered by current shape-based retrieval approaches without any user intervention. In this paper, we present a new cooperative approach based on the shape neighbourhood 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.

Keywords

Visual information retrieval, Shape-based query rewriting.

Received January 19, 2006 / Accepted March 14, 2006

1. Introduction

Visual query languages have been the focus of many studies [27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42] and have been used in several domains like Geographic Information Systems, Medical images, Digital museums, time series applications, etc. A visual language is based on pictorial representation of conceptual entities and operations through which users compose iconic or visual sentences [41]. 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 [27, 31, 33], Sketch-based [32, 35], and Query By Image [37, 42]. 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 [33]. In sketch languages, the queries are user-talent dependent and may lead to several interpretations [33]. 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 [12, 13].

Widely used in several search engines and for textual data, the query rewriting technique has been studied in several domains [22]. The relevance feedback is one of the query rewriting techniques [38, 39, 40]. 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 [22], 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:

1. Including and/or excluding shapes from the relaxation result,
2. Assigning an order to relaxation results according to the user requests.

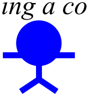
This paper is organized as follows. First, we explain the motivation of this work. After, we give a snapshot of the related work. In section 4, we detail our rewriting method, and give some examples. Section 5 is devoted to present our implementation. Finally, we conclude and pin down some of our future directions.

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 1*) 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*

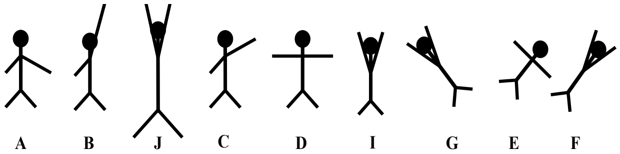


Figure 1: A set of shapes representation in the database

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

To obtain expected results, the following steps are applied:

1. *The journalist gives the initial query Q (shape D) using a threshold ε_1 (we assume here that the distance within $\varepsilon_{1,}$ 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*

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

a. *shape D and J (which are expected)*

b. *shape F and G (close to I) which are unexpected*

5. *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.

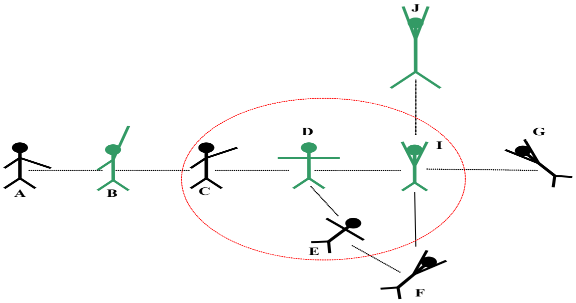


Figure 2: Similarity links between shapes

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*: considered as being a valuable tool to improve the performance of retrieval systems [3, 9], they are used to obtain a new query by modifying the initial query on the basis of current user feedbacks. The relevance feedback[2] is one example of such used techniques.
- *User-oriented techniques*: rewrite the initial query according to predefined parameters stored in a user model or profile [43] without studying the current needs of the user. Rewriting parameters can be based on user IP address, language, country, etc.

Below, we snapshot current visual and shape-based query languages and their related rewriting techniques.

3.1 Iconic Languages

Iconic languages allow the user to formulate queries using predefined icons representing domain-related objects and operators. In [27], the authors present a visual query language, MQuery, able to support multimedia data in addition to alphanumeric one. Using MQuery, the user formulates his query by dragging shape elements from the schema diagram to a query window and asks the system to *insert, retrieve, delete, or update* any matching data. In [31], CIGALES language allows the user to formulate queries using predefined icons. Icons are able to express spatial objects and several operators like *inclusion, intersection, adjacency, etc.* In [33], the iconic language LVIS, is presented and allows more flexibility concerning the ambiguity resolution in complex queries.

3.2 Sketch Languages

Using sketch languages, the user can formulate his query without the constraints of predefined icons. In [32], using the *Sketch!* language, the user formulates his query by drawing spatial objects and operators. The main contribution in [35], is to make a user interface (UI) be more *Humanistic, Intelligent,* and *individualized*. Two ways of input sketch are defined. The first way is the off line case where the sketch is given as a query image and analysed to extract drawn components, whereas the second way is the on line case in which the system gives dynamic interpretation to the

[2] We will give more details about it in the following subsections.

user and provides him the possibility of adjusting the result. For example, if the user has the intention of drawing an equilateral triangle, the UI corrects the inexact drawn triangle. In addition, input sketch for the same shape may be different from user to user. The authors construct a multi-class classifier using learning algorithms to deal with user adaptation.

3.3 Query by Image Language

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 [37], the authors describe a visual query language (VQL) among time series data. Here, the user interactively specifies a search pattern, after what the system finds similar shapes and returns a ranked list of matched ones. In addition, the user is able to specify a set of parameter series including minimum fitness of the search, expected periodicity, and number of hits to return. Another interesting shape application on time series and life time data is presented in [42], where the authors defined the similarity transformation distance, in order to measure the similarity between a shape and its transformation within a given threshold.

3.4 Relevance Feedback Techniques

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 [38], a survey on relevance feedback in CBIR is presented. The central idea in classical methods of relevance feedback is query re-weighting. The re-weighting aims at enhancing the importance of those dimensions of a feature that helps in retrieving and reduces the importance of those that hinder this process. However, low-level features alone are not enough effective in representing users' feedbacks and in describing their intentions. For this reason, the authors present *iFind,* a web-based image retrieval system that combines keyword-based image search and query by image technique. In [39], 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. Using this list, they define training data, employ a technique of learning from this learning data, infer a concept form this data, and produce other instances from a database that are consistent with this concept. Another interesting work is addressed in [40], where the authors present three levels: the object, the physical features and the representation of each feature (for example color histogram and color moment for the color feature). Each level is expressed by a vector having several components. To integrate relevance feedback, the authors add a set of similarity measures to each feature representation, and assign weights to each component in the three layer object/query representation. Initially, the weights at each level are equal. After each iteration of the image retrieval algorithm, the system returns a set of images after what the user assigns a score reflecting the relevance of the retrieved ones. Using this score, the algorithm updates weights and recalculates similarities.

3.5 Limits

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, these techniques did not deal with the following issues:

- 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 which might be of great importance in shape-based retrieval because priorities would express the real word significance of each shape.
- 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 Rewritng

In this paper, we extend the textual-oriented rewriting approach presented in [22] 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[3]. After, we define the concept of shape neighbourhood. The cost and the neighbourhoods are defined with respect to each shape transformation. To classify neighbourhood 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.

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 [1,6,7,14,15,17] and graph-based approaches [5,11,18,19,20,21]. For instance, in graph-based approach [20], 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 follows:

- Contour matching: considering a_1, a_2 two nodes in the graph of A and b_1, b_2 their mapping nodes in the graph of B. The cost between edges (a_1, a_2) and (b_1, b_2) is calculated from the cost of comparing their correspondent contours. The calculation function returns the cost with respect to the curvatures at (a_1, a_2) and (b_1, b_2).
- Transformation matching: the structure of a graph may change due to the transformations like *stretching and occlusion*. A transformation cost is calculated for each transformation. For instance, the stretching transformation cost is calculated by comparing an edge of a shape graph with a path of the other shape graph. Graph matching operations can be done using the A*LIKE algorithm [43].

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

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

Definition 2: Shape Neighborhood

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 $\varepsilon_{R.}$ 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:

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

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:

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

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

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

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 [22]. They allow the user to include or exclude, from the query, using a given threshold and according to his interest and needs. In this paper, we extend their use. We apply these two function s not only on textual terms and relations but also on shapes. We detailed the extension 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:

$$\mathbf{Rewrite(shape\ element) = Rewrite(\delta) = F_C(F_R(\delta)) = F_C(\delta') = \delta'_C}$$

We detail each concept below:

Definition 4: Shape element δ

The shape element **δ** is a triplet ($\mathcal{A}$, $\mathcal{R}$, $\mathcal{T}$) 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 $\mathcal{A}$ in R. $\mathcal{T} \in [0, 1]$ and represents the maximum distance of a shape B $\in V_R$ ($\mathcal{A}$) to consider in the relaxation.

Definition 5: Relax function F_R

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 $\mathcal{A}$ in descending order. Each couple of **δ'** is a node (value and weight) selected from the neighbourhood of $\mathcal{A}$ with respect to the transformation $\mathcal{R}$. The distance (weight difference) between $\mathcal{A}$ and a selected shape is less than or equal to $\mathcal{T}$. $\mathbf{F_R}$ is formally formulated as follows:

$\mathbf{F_R: \delta = (\mathcal{A}, \mathcal{R}, \mathcal{T}) \rightarrow F_R(\delta) = \delta' = \{(shape, W), \leq_W\}.}$ Where $\leq_W$ indicates that δ' is an ordered set according to 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:

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

$$\cup$$

$$\mathbf{\{(\delta_p', W) \text{ where } (\delta_p', W) \in C \text{ and } 0 < W \leq 1\}}$$

In other words:

- If W = 1 then $\mathbf{F_C}$ includes the shapes of $\mathbf{\delta_p'}$ to the relaxation result **δ'**,
- If W = 0 then $\mathbf{F_C}$ excludes the shapes of $\mathbf{\delta_p'}$ from relaxation result **δ'**,
- If 0 < W < 1 then $\mathbf{F_C}$ modifies the weight of the shapes of $\mathbf{\delta_p'}$ 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 $\varepsilon \in [0,1]$ 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*
    a. *exclude rotated shapes* (ε_R=0)
    b. *include occluded shapes using* ε_O
    c. *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*

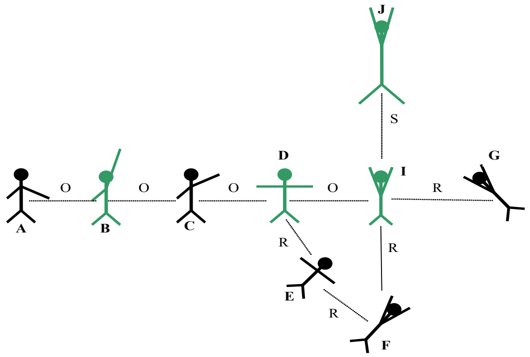


Figure 3: Tranformation links between shapes

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 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 [1,6,7,14,15,17] and graph-based approaches [5,11,18,19,20,21] are provided in the literature to represent the shape. In our prototype, we adopted a graph-based method called ***shape-axis*** [19,20,21]. It consists of representing each shape by a unique axis tree for similarity computation. It has been chosen for various reasons:

- The shape-axis method can illustrate the skeleton of both the open and the closed curves as mentioned in [19], because its main goal is to determine the axis that describes the shape independently of the continuity or the discontinuity of the curve. The major handicap of the segmentation method, studied in [4], is the absence of considering an open curve to determine its unique skeleton representation. This can present a real problem when trying to treat such a contour.
- The *shape-axis* representation is sensitive to stretching; this can be solved by merging some correspondent nodes together. It presents also some sensitivity toward the occlusion that can be treated by cutting some unused nodes. These operations are used to minimize the similarity distance between two shapes. In [3], these operations are studied and the authors try to calculate the cost of similarity between two shapes.

To match two shapes, we only need to match their corresponding graphs. The graph matching problem has been widely studied in the literature [24, 25]. Basically, a graph representation **G= (V, E)** is composed by a set of nodes V and a set of edges E. **Exact** and **inexact** graph matching are two basic types of graph matching (isomorphism and homomorphism respectively). The best correspondence of a graph matching problem is defined as the optimum of some objective function which measures the similarity between matched vertices and edges. This objective function is also called *fitness function* [24]. In other words, this function measures the fitness of the isomorphism and the homomorphism defined between the two graphs. In shape axis method, the A*LIKE algorithm is used to match graphs of two different shapes, and calculated costs are used to measure the dissimilarity of the matched graphs.

In the following subsections, we briefly explain how the cost of two graphs matching is calculated in the case of stretching transformation[4], and show a running retrieval example using our prototype.

5.1 Neighborhood and relaxation

Based on matching cost, the shape neighborhood is defined according to the stretching transformation allowing to apply relaxation and constraint functions used in our approach.

As mentioned and studied in [20], we specify for each shape a unique SA-tree. The SA-tree represents the axis of the shape. Stretching or articulating a shape causes some deformations, and consequently changes its appropriate shape representation, by adding additional bifurcations (internal segments connecting two nodes).

[4] Other transformations are computed similarly.

Referring to Figure 4, we observe the occurred changes that increase the complexity of calculating the similarity between two inexact matching shapes due here to the added segment (bifurcation between the nodes [B_2,B_3]) on the articulated image (*shape B*).

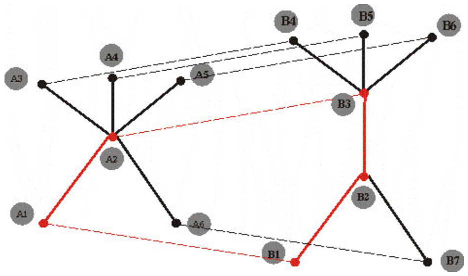


Figure 4:The graph of representation of two different shapes

The similarity between A and B can be identified by computing the correspondence (commonly called Edge-to-path correspondence) between them using a merging operation. In [20], the authors tried to find the total cost of correspondent segments Cost_S, and suggested to use a penalty cost for the merging operation. This cost is called Cost_M 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. To calculate the cost between the edge (A_2, A_1) and the path (B_1, [B_2, B_3]), we must take into account the cost of the correspondent edges (B_2, B_1), (A_2, A_1), with addition to the merging cost Cost_M as follows:

$$\text{Cost}[\text{p}(B_1,[B_2,\ B_3]),e(A_2,\ A_1)]\ =\ Cost_S(CT(B_2,B_1),CT(A_2,\ A_1))+Cost_M([B_2,\ B_3],\ A_2)$$

Where:

- **B**: is an image with additional segment on its representation (containing two bifurcation nodes).
- **A**: original image
- **[B_2, B_3]**: is the additional edge to be merged

The correspondence in this case is edge-to-path correspondence, where the edge **e(A_2, A_1)** corresponds to the path **p(B_1, B_2, B_3)**.

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

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

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

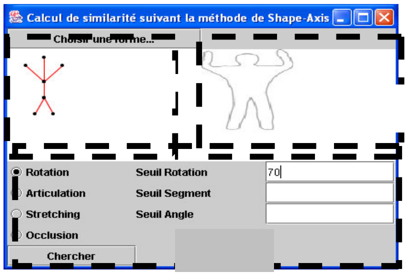
$$V_S(A,\ C_s) = \{B\ /\ (\text{Cost}_S(A,B)+\text{Cost}_M(A,B)\ \leq\ C_s\ )$$

V_S represents the neighborhood set that enables 2 parameters:

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

6. Prototype

Our java-based prototype uses a shape-based visual retrieval interface (Figure 5) composed of several zones:



● Rotation Seuil Rotation 70

○ Articulation Seuil Segment

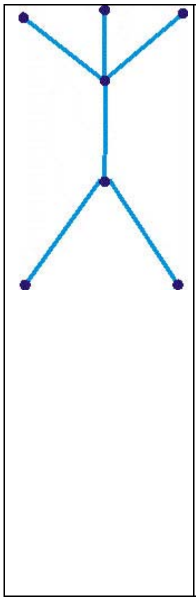
○ Stretching Seuil Angle

○ Occlusion

Figure 5:Prototype retrieval interface

1. **Zone 1**: used to formulate a visual query by selecting an existing image or by drawing a new one. This version of our prototype allows the users to only select SVG documents.
2. **Zone 2**: contains the corresponding graph of the shape given by the user. We use XML (eXtensible Markup Language) to represent the shape graph structure. Figure 6 shows an example of the XML file representing the shape axis of the initial shape given by the user.
3. **Zone 3**: allows the user to include or exclude one or more of the following transformations:
   - *Rotation* with a rotation threshold
   - *Occlusion* with segment and/or angle variations thresholds
   - *Articulation* with segment and/or angle variations thresholds
   - *Stretching* with segment variation threshold

According to the input shape query and parameters given by the user, the prototype computes the shape neighbourhoods (one per transformation), and returns the result of rewriting process.



```
<?xml
version="1.0"encoding="U
TF-8" ?>
<shape-axis>
 <edge id="0">
  <pointD x="30" y="80" />
      <pointA      x="40"
y="60" />
 </edge>
   <edge id="1">
      <pointD      x="40"
y="60" />
      <pointA      x="40"
y="40" />
  </edge>
 </shape-axis>
```

Figure 6:The shape axis of the initial shape and its corresponding XML file

If we consider the running example in Figure 7, the prototype shows the visual query rewriting result after the unique rotation inclusion, with a given threshold of 70.

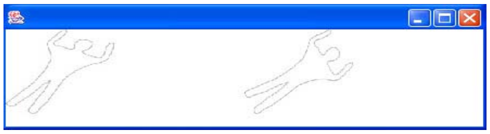


Figure 7: Results with only rotation

The user can also select several independent transformations with various corresponding thresholds. Figure 8 shows the query rewriting results, where the user includes occlusion and stretching transformations and excludes other transformations.

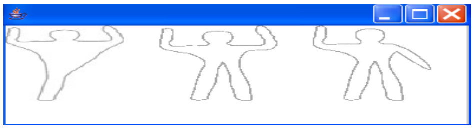


Figure 8: Results with only occlusion and stretching

7. 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.

8. References

[1] 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

[2] Sundar, H., Silver, D., Gagvani, N. and Dickinson, S. *Skeleton Based Shape Matching and Retrieval*. Shape Modeling International, pp. 130-139, 2003

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

[4] Shah, J. *Skeletons of 3D Shapes*. 5th international conference on Scale Space and PDE methods in computer vision. 339-350 2005

[5] Kupeev, K. and Wolfson H. *On Shape Similarity*. Computer Vision and Image Processing 227-231, 1994

[6] Latecki, L. J., Lakaemper, R. and Wolter, D. *Optimal Partial Shape Similarity*. Image and vision computing, 227-236, 2005

[7] Carlin, M. *Measuring the Performance of Shape Similarity Retrieval Methods*. SINTEF Electronics and cybernetics, vol.84 (1), pp. 44-61, Norway, 2001.

[8] "*Maxima and Minima*". http://www.mathcentre.ac.uk, MathCentre, August, 2004.

[9] Ortega-Binderberger, M., Mehrotra S. *Relevance Feedback in Multimedia Databases*, Handbook of Video Databases: Design and Applications", CRC Press, 2003

[10] Remco C. Veltkam and Hagedoorn, M. Shape Similarity Measures, Properties and Constructions. 4th International Conference on Advances in Visual Information Systems, 200: 467 – 476, 2000.

[11] Tam, R., Heidrich, W. *Shape simplification Based on the Medial Axis Transform*. 17th conference Eurographics, pp. 481-488, 2003.

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

[13] Gaasterland, T., Godfrey, P. and Minker, J. *Relaxation as a platform of cooperative answering*. Journal of Intelligent Information Systems, pp 293-321, 1992.

[14] Basri, R., Costa, L., Geiger, D. and Jacobs, D. *Determining the similarity of deformable shapes*. Vision Research 38: 2365 – 2385, 1998

[15] Frese, T., Bouman, A. C., and Allebach, J. *A Methodology for Designing Image Similarity Metrics Based on Human Vision System Models* Tech. Rep. Purdue University, West Lafayette, 1997

[16] Thomas, S., Bejamin, K. *Curves vs. Skeletons in Object Recognition*. Proceedings IEEE International Conference on Image Processing, pp 22-25, 2001.

[17] Adamek, T., O'Connor, N. *Efficient Contour-based Shape Representation and Matching*. Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval, 2003.

[18] Tung, T., Schmitt, F. *Indexation d'objets 3D par Graphe de Reeb Multi résolution Augmenté*. International Journal of Shape Modeling (IJSM), 11(1): 91-120, June 2005

[19] Tyng-Luh Liu. "*A Generalized Shape Axis Model for Planar Shapes*". IPCR, pp. 3491-3495, 2000.

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

[21] Liu, T., Geiger, D. and Kohn R. *Representation and Self-Similarity of Shapes*". Int'l Conf. Computer Vision, Bombay, 1129-1138, 1998

[22] Chalhoub G., Saad S., Chbeir R. and Yetongnon K., "*Adaptive data retrieval in multimedia DBMS*", proceedings of the CSITeA-04 conference, December 27-29, Cairo, Egypt, 2004

[23] Hagedoorn, M., Veltkamp, R, Measuring Resemblance of Complex Patterns , Lecture Notes in Computer Science, 1568: 286-298, 1999

[24] Bengoetxea, E., *Inexact Graph Matching Using Estimation of Distribution Algorithms* , Phd thesis Ecole Nationale Supérieure des Télécommunications (Paris) 2002

[25] Bengoetxea, E., "*Graph Matching as a combinatorial Optimization Problem With Constraints*", Ecole Nationale Supérieure des Télécommunications (Paris) 2002.

[26] Chang, S. K., Orefice, S., Tucci, M. and Polese, G. *A Methodology and Interactive Environment for Iconic Language Design*, International Journal of Human (Computer Studies), 41(3): 683-716, 1994

[27] Dionisio, J., Cardenas, A. MQuery: *A Visual Query Language for Multimedia, Timeline and Simulation Data*, Journal of Visual Languages and Computing 1996, 377-401

[28] Smith, J. R., Chang, S.F. "*Visually Searching the Web for Content,"* IEEE Multimedia Magazine, 1997, Vol. 4, 3, 12-20.

[29] *VideoQ*, http://persia.ee.columbia.edu:8080/visual_help.html (last visited on 30/08/2005)

[30] M. Beigi, A. Benitez, S.-F. Chang, "*MetaSEEk: A Content-Based Meta Search Engine for Images,"* SPIE Conference on Storage and Retrieval for Image and Video Database, San Jose, pp118-128 Feb. 1998.

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

[32] 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, pages 113-135, 1992.

[33] Aufaure, M. , Bonhomme, C. , and Lbath, A. LVIS : *Un Langage Visuel d'Interrogation de Bases de Données* Spatiales , BDA'98, Tunisie, pp 527-545, 1998

[34] Di Sciascio, E. and Mongiello, *M. Query by sketch and relevance feedback for content-based image retrieval over the web*. Journal of Visual Languages and Computing, 10(6):565-584, 1999

[35] Xiaogang X., Wenyin L., Xiangyu J., and Zhengxing S. *Sketch-based user interface for creative tasks*, In Proc. 5th Asia Pacific Conference on Computer Human Interaction HI2002), pages 560-570, 2002.

[36] Leifman, G., Meir, R. and Tal, A., *Relevance Feedback for 3D Shape Retrieval,* The 5th Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics, pages: 15-19, 2004.

[37] Haigh, K., Foslien, W., and Guralnik, *V. Visual Query Language: Finding patterns in and relationships among time series data*. Seventh Workshop on Mining Scientific and Engineering Datasets, 24 Apr 2004

[38] Zhang, H., Chen, Z., Liu, W. *Relevance Feedback in Content-Based Image Search*. ww.research.microsoft.com

[39] 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, June 2000.

[40] Rui, Y., Huang, T., Ortega, M., and Mehrotra, S. "*Relvance feedback: A power tool for interactive content-based image retieval"* IEEE transactions on circuits and video technology 1998

[41] Chang, S., Costagliola, G., Pacini, G. and Tucci. M., "*Visual Language System for User Interfaces*," IEEE Software, pages. 33-44, March 1995.

[42] Goldin D. et al. *Normalization of Life Science Data for Shape-based Similarity Querying* BECAT/CSE Technical Report TR-04-1, January 2004

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

[44] 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.