Erratum

In the printed book pages of the paper “Vasilis Delis, Dimitris Papadias: Querying Multimedia Documents by Spatiotemporal Structure, LNAI 1495, pp. 126-138, 1998” were exchanged. On the following pages you can find the correct article.

Querying Multimedia Documents by Spatiotemporal Structure

Vasilis Delis[1], Dimitris Papadias[2]

[1]Computer Engineering and Informatics Department, University of Patras, Computer Technology Institute, Kolokotroni 3, 26221, Patras, Greece - delis@cti.gr
[2]Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong - dimitris@cs.ust.hk

Abstract. Interactive multimedia documents are rapidly becoming available on the WWW. Navigating in such large document repositories requires flexible retrieval techniques based on spatiotemporal structure. In this paper we address this issue by extending the notion of *conceptual neighborhood* (a concept describing similarity among relations between time intervals) to various resolution levels and higher dimensions. We propose a binary string encoding of relations which allows the automatic derivation of similarity measures and apply our framework for effective support of multimedia queries.

1 Introduction

There is nowadays an abundance of multimedia documents on the WWW, from simple HTML pages to complex multimedia presentations. Current IT users are confronted with huge amounts of information, originating from various sources and containing several forms of data ranging from text to sounds, slide shows etc. In the sequel, the term "multimedia document" (or simply document) will be overloaded to refer to any kind of the above information structures, presented on a computer screen. The specification of such a document entails definitions of its *content* (semantic information), *interactivity* (control flow based on user interaction) and *structure* (which refers to spatiotemporal relations among the document's objects).

Research conducted in the information retrieval field so far mainly focuses on content-based retrieval, covering many types of data forms, like text [13], images [4] and video [14]. However there is an increasingly evident demand[1] for a new paradigm for querying and navigating in the available document repositories: users should be allowed to ask for documents based on their structure and metainformation, in addition

[1] This paradigm shift in information retrieval involves a series of research steps, from building conceptual multimedia models able to capture structure, to developing intelligent search engines able to retrieve documents according to structure and metainformation. The importance of handling this type of advanced information can be also demonstrated by the recent efforts to establish sophisticated mark-up languages like SGML and XML.

T. Andreasen, H. Christiansen, H.L. Larsen (Eds.): FQAS'98, LNAI 1495, pp. 126-138, 1998.

to content [7]. For example, a traveller may be interested in getting all multimedia documents containing images of hotels *immediately followed* by a slide show, or a car buyer may want to search for a document containing a 3D video demo of a Ferrari's interior *adjacent to* a text window *on its left*.

The processing of such queries should be sufficiently flexible to allow partial matches because the difference between objects that satisfy the query, and the ones that don't, may be quantifiable and gradual. Our work proposes a framework for similarity retrieval of multimedia documents, and does so by touching upon two issues: we first establish a formal framework for the definition and representation of spatio-temporal relations and show how structural relation-based similarity can be effectively accommodated in this framework. The rest of the paper is organized as follows: Section 2 introduces a new encoding of relations which facilitates conceptual neighborhood inference and its extensions to multiple resolution levels and dimensions. Section 3 applies the framework for the formulation of spatiotemporal queries on multimedia documents. We conclude in Section 4 by outlining future continuation of this work.

2 Representing Relations in Multimedia Documents

Relation neighborhoods constitute an effective way to deal with spatiotemporal queries. Freksa [5] defined the concept of *conceptual neighborhood* as a cognitively plausible way to measure similarity among Allen's [1] interval relations. A neighborhood is represented as a graph whose nodes denote *primitive* relations that are linked through an edge, if they can be directly transformed to each other by continuous interval deformations.

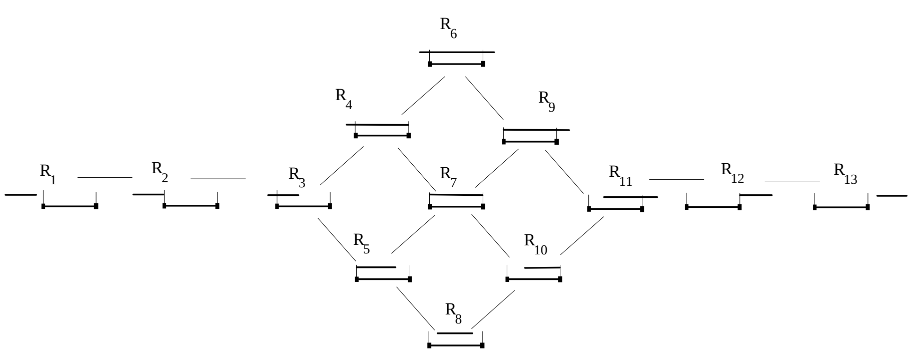


Fig. 1. Conceptual Neighborhood for relations between intervals in 1D space

Depending on the allowed deformation (e.g., movement, enlargement), several graphs may be obtained. The one in Figure 1, corresponds to what Freksa called *A-neighbors* (three fixed endpoints while the fourth is allowed to move). Starting from relation R_1 and extending the upper interval to the right, we derive relation R_2. With a similar extension we can produce the transition from R_2 to R_3 and so on. R_1 and R_3 are called 1st degree neighbors of R_2. Although hereafter we assume type A neighborhoods, extensions to other types are straightforward.

First we introduce a few definitions, to be used later in the paper. A *relation set r*, represents a disjunction of primitive relations. The distance between a relation set *r* and a primitive relation *R* is the minimum distance between any relation of the relation set and *R*, i.e. $d(r,R) = min(R_k,R)$, $R_k \in R$. We will also define the additional relation R_U which denotes the *universal* relation (the disjunction of all relations); it is used in queries to leave the relation between two objects unspecified.

Fig. 2. Encoding of spatial relations

In order to provide a general framework for relation-based similarity, in the sequel we propose a new encoding of relations. Consider a (reference) interval [a,b]. We identify 5 distinct 1D regions of interest with respect to [a,b]: 1.(-∞,a) 2.[a,a] 3.(a,b) 4.[b,b] 5.(b,+∞). The relationship between a (primary) interval [z,y] and [a,b] can be uniquely determined by considering the 5 empty or non-empty intersections of [z,y] with each of the 5 afore-mentioned regions, modelled by the 5 binary variables *t, u, v, w, x*, respectively, with the obvious semantics ("0" corresponds to an empty intersection while "1" corresponds to a non-empty one). Therefore, we can define relations in 1D to be binary 5-tuples (R_{tuvwx} : $t, u, v, w, x \in \{0,1\}$). For example, R_{00011} (t=0, u=0, v=0, w=1, x=1) corresponds to the relation of Figure 2 (R_{12} in Figure 1). A binary tuple represents a valid spatial relation if it contains a list of consecutive "1"s (in case of a single "1", this should not correspond to u or w, otherwise [z,y] collapses to a point) and the intervals of interest form a consecutive partition of (-∞,+∞).

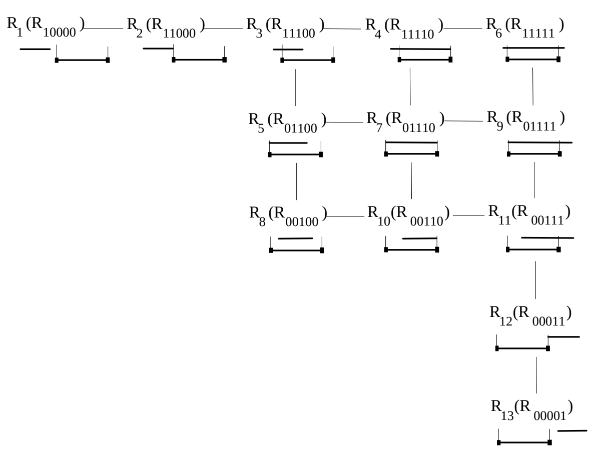


Fig. 3. A rearrangement of the 1D neighborhood

The new notation is more expressive, in the sense that given a relation, the user can easily infer the corresponding spatial configuration and vice versa (the only means to achieve this using the former notation is by referring to the neighborhood graph). In Figure 3 we present a correspondingly more expressive re-arrangement of the conceptual neighborhood graph.

The edges of the graph are arranged horizontally and vertically and the semantics of traversing the graph in either direction are captured by the following "pump and prune" rule of thumb: Given a relation R_x, there are 4 potential neighbors, denoted $right(R_x)$, $left(R_x)$, $up(R_x)$, $down(R_x)$, respectively, with the obvious topological arrangement in the graph. $right(R_x)$ can be derived from R_x by "pumping" an "1" from the right, i.e. finding the first "0" after the rightmost "1" and replacing it by a "1". $left(R_x)$ can be derived from R_x, by "pruning" an '1" from the right, i.e. replacing the rightmost "1" by a "0". Similarly, $up(R_x)$ can be derived from R_x by pumping an "1" from the left while $down(R_x)$ can be derived by pruning the leftmost "1".

Notice that not all neighboring relations are always defined. Consider, for example, the relation R_{11000}, for which $up(R_{11000})$ is not defined since the leftmost digit is a "1" while $down(R_{11000})=R_{01000}$ is not a valid relation. Let n be the number of bits used to encode relations (n=5 for Allen's relations). Assuming that each relation is a string of bits where the nth is the rightmost, the following pseudo code computes the right neighbor of a relation. The other neighbors are computed in the same way.

```
Right_Neighbor(relation R) {
   i:=rightmost_1(R); //calculates position of rightmost "1"
   if (i=n) return ∅
     else return (set_bit(R,i+1))
}
```

Since movement in the neighborhood graph is restricted to horizontal and vertical directions, the distance between any two nodes(relations) is the sum of their vertical and horizontal distances. Thinking of it intuitively, the distance between any two relations can be calculated by counting how many elementary movements we have to perform on an interval in order for the two relations to become identical. The larger the number of simple movements, the less similar the relations. The binary string representation enables the simplification of the previous procedure to the following one: we only have to compare the notation of the two relations and count the minimal number of "1"s that we have to add in order to make the two notations two identical binary strings in which the "1"s are consecutive. The corresponding pseudo code is given below:

```
distance(relation R1, relation R2) {
   R = R1 OR R2;  //bitwise OR
   count = 0;
   for i:= leftmost_1(R) to rightmost_1(R) {
      if R1[i]=0 then count++;
      if R2[i]=0 then count++;
    } //end-for
   return(count);
}
```

For example $d(R_{\underline{000}11}, R_{1\underline{0000}}) = 7$ and $d(R_{\underline{0}11\underline{0}0}, R_{11110}) = 2$ (the underlined 0s are the ones counted during the calculation of distance). The above ideas can be extended in order to handle relations at varying resolution levels, while retaining all the good properties (expressiveness, inference of neighboring relations, easy distance calculation). We will initially illustrate the applicability of "binary string" encoding to a coarse resolution level, i.e. to a level where only a few relations can be distinguished. In the example of Figure 4, the 1D regions of interest are (-∞,a), [a,b] and (b,+∞), respectively. The corresponding relations are of the form R_{tuv}, $t,u,v \in \{0,1\}$. This allows for the definition of only 6 primitive relations since information content concerning the endpoints of [a,b] is reduced: R_{100} (*before*), R_{010} (*during*), R_{001} (*after*), R_{110} (*before_overlap*), R_{011} (*after_overlap*), R_{111} (*includes*). Figure 4 illustrates four configurations that correspond to R_{010} and cannot be distinguished in this resolution.

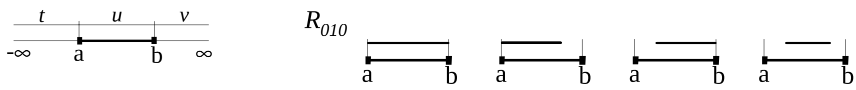


Fig. 4. Encoding at a coarse resolution level

Increasing the resolution of relations in our model can be achieved simply by augmenting the number of regions of interest (the number of bits in the binary string representation), thus refining the information level for a particular spatial relationship. Figure 5 illustrates the complete conceptual neighborhood graph for distance relations that can be defined using two additional points outside the reference interval (the number of required bits is 9).

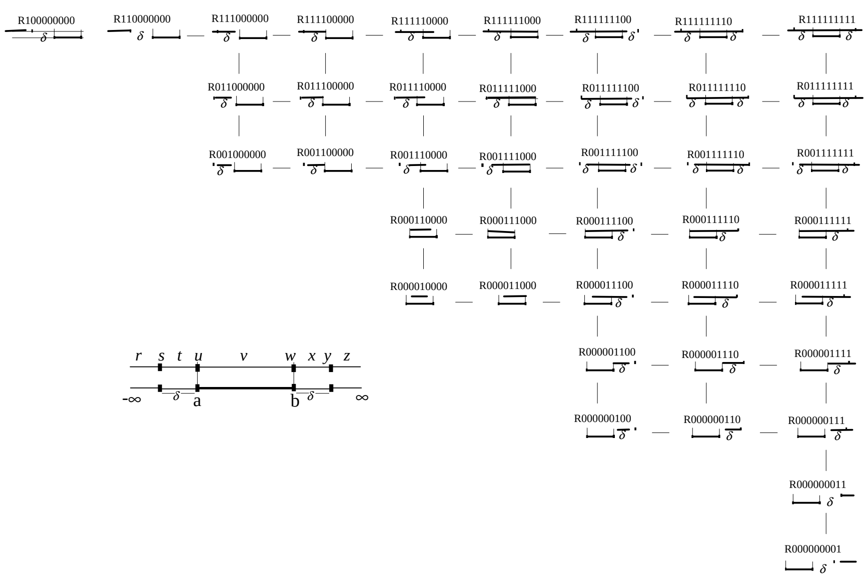


Fig. 5. 1D Conceptual neighborhood including distances

In this framework, *left_far*, for instance, can be defined as $R_{100000000}$. An arbitrary number of, possibly different, interval-extensions can be used to define as many relations as needed to match the application needs. We call such consecutive partitionings of space *resolution schemes*. In general, if n is the number of bits used by the resolution scheme, the number of feasible relations in 1D is $n(n+1)/2 - k$, where k is the number of bits assigned to single points (i.e. intervals of the form [a,a]). If we fix the starting point at some bit then we can put the ending point at the same or some subsequent bit. There are n choices if we fix the first point to the leftmost bit, n-1 if we fix it to the second from the left, and so on. The total number is $n(n+1)/2$ from which we subtract the k single-point intervals. For the 1st scheme ($n=5$, $k=2$) we get 13 (Allen's) relations, while for ($n=9$, $k=4$) we get the 41 relations of Figure 5.

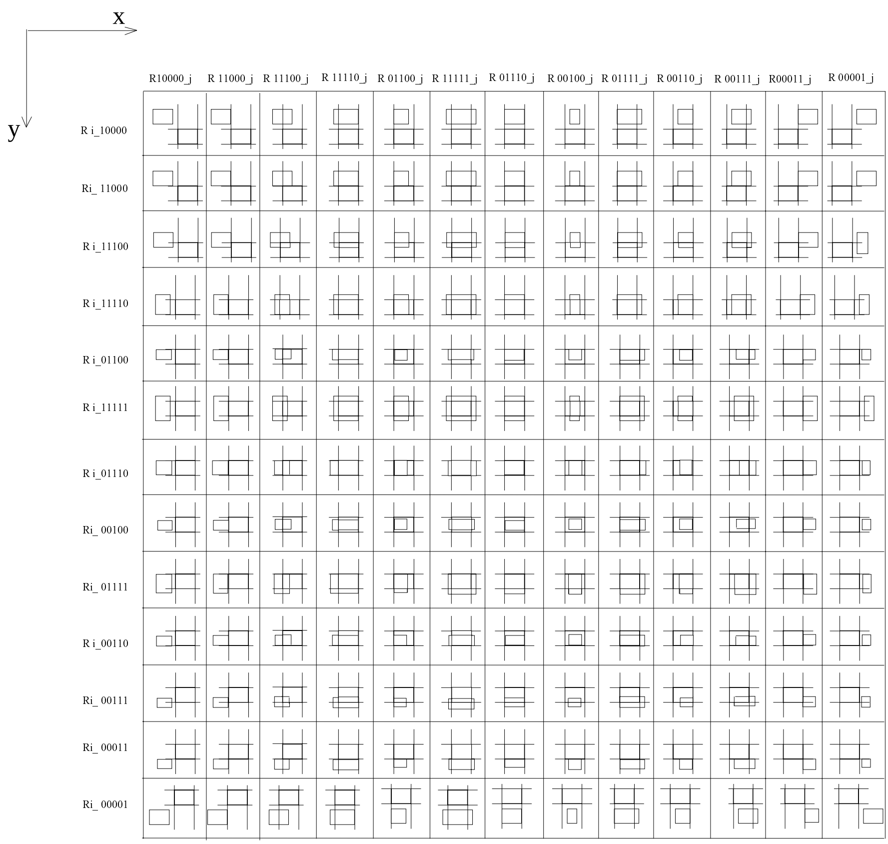


Fig. 6. Relations between 2D regions

A N-dimensional projection relation is a N-tuple of 1D relations, e.g. $R_{11000\text{-}11100}$ = (R_{11000},R_{11100}). Each 1D relation corresponds to the relationship between the N-dimensional objects in one of the dimensions. So if s is the number of possible 1D relations at a particular resolution, the number of ND relations that can be defined at the same resolution is s^N. According to the requirements of the particular application, not all dimensions need to be tuned at the same resolution, in which case the maximum number of ND relations is the product of the corresponding numbers for each dimension. Figure 6 illustrates the 169 (13^2) primitive projection relations between regions on the plane, at the initially discussed (Allen's) resolution scheme.

All previous properties can be analogously extended to N dimensions. So, given a N-dimensional relation, the corresponding spatial configuration can be easily inferred by combining all the 1D configurational inferences. In Figure 6, each column corresponds to a spatial configuration on the x axis. As we move along a column, we get all the possible y axis variations of this configuration.

Since in one dimension a relation may have 4 potential neighbors, in N dimensions every relation has a maximum of 4N neighboring relations. Consider an N-dimensional relation $R_{i1i2\ldots iN}$ where each i_k is a binary string. In order to derive a neighboring relation we have to replace one of its constituent 1D relations R_{ik} with its 1D neighbors, say R_{xkj}, j=1..4, i.e. $neighbor(R_{i1i2\ldots ik\ldots iN}) \in \{R_{i1i2\ldots xkj\ldots iN} | j=1..4, k=1..N\}$.

As a result, computing *ND* neighbors is reduced to the already solved problem of computing 1D neighbors, thus the "pump and prune" method can still be applied to construct the conceptual neighborhood graph. Assuming the block world metric, the distance between two *ND* relations is the sum of the pair-wise distances between the corresponding constituent 1D relations, i.e. $d(R_{i1i2\ldots iN}, R_{j1j2\ldots jN}) = d(R_{i1},R_{j1}) + d(R_{i2},R_{j2}) +\ldots+ d(R_{iN},R_{jN})$.

The advantages of the proposed encoding is that it uniformly treats all types of (spatial and temporal) relations in various resolution levels and dimensions and inference, like conceptual distance calculation and conceptual neighborhood derivation, can be fully and efficiently automated, while the algorithms for neighborhood computations and retrieval remain the same. [8] and [2] apply conceptual neighborhoods for configuration similarity retrieval in GIS. Unlike the proposed methods, the above techniques do not uniformly treat all types of spatial relations, while they assume fixed domains and queries, i.e., the permitted relations (direction and topological) are defined in advance and can't be tuned to different resolutions. In the following section we show how our framework can be adapted for similarity assessment among multimedia documents.

3 Multimedia Similarity Queries

A multimedia document can be thought of as a collection of 3-dimensional objects: a series of 2D snapshots along the time axis in which multimedia components (buttons, images, etc.) are "on" for certain periods of time. Conceptual neighborhoods for projection-based definitions of relations are particularly suitable for structural similarity retrieval:

- most often in practice multimedia documents consist of rectilinear objects (e.g., window objects). Projection-based relations provide an accurate and effective means for spatio-temporal representation of collections of such objects [10].

– even for non-rectilinear objects, usually spatial/multimedia databases utilise minimum bounding rectangles (MBRs) for efficient indexing. MBRs provide a fast *filter*[2] *step* which excludes the objects that could not possibly satisfy a given query. The actual representations of the remaining objects are then passed through a (computationally expensive) *refinement step* which identifies the actual results.
– multimedia queries do not always have exact matches and crisp results. Rather, the output documents should have an associated "score"[3] to indicate the similarity between their retrieved spatiotemporal relations and the target relations of the query document. By adoption, this score is inversely proportional to the degree of neighborhood.

Multiresolution and multidimensional conceptual neighborhoods can be tuned to provide flexible query answering mechanisms. For instance, in a particular situation where refined queries involving distances are needed, a resolution scheme such as the one depicted in Figure 5 (or even with multiple distance extensions) should be used. The conceptual multimedia model and query language must be extended accordingly to allow the expression of relevant queries (e.g. inclusion of keywords such as *left, near,* etc.) and the retrieval algorithms must be adjusted for efficient retrieval. For the sake of simplicity, the following examples will be based on Allen's relations (the first discussed resolution scheme).

Assume the following situation. A user is searching a tourist multimedia database and recalls having browsed a document with a structure described in the following query: "find all tourist documents in which five-star hotels are described by a background image *i*, which *contains* a textual description *t*, *immediately followed* by a corresponding video clip *v* on the *right* of the text window". The spatiotemporal relations of the query document are $R_{11111\text{-}11111\text{-}time}(i,t)$, $R_{11111\text{-}11111\text{-}time}(i,v)$, $R_{X\text{-}U\text{-}11000}(t,v)$ (assuming the order is: the first two strings for *x* and *y* axes, respectively, and the third for the *time* axis). Notice that the *y* axis relation between the video clip and the text window is unspecified in the query statement, thus defined as R_U, while the following mappings hold:

$R_{time}(i,t)\rightarrow R_{11111}(i,t)\vee R_{01111}(i,t)$, $R_{time}(i,v)\rightarrow R_{11111}(i,v)\vee R_{11110}(i,v)$,
$R_X(t,v)\rightarrow R_{11000}(t,v)\vee R_{10000}(t,v)$

This query essentially describes a spatiotemporal scene, so it constitutes a multimedia document itself. Unlike stored documents where all the relations between all pairs of objects are explicitly represented, query documents may be *incomplete* (the relation between a pair of objects may be left unspecified), *indefinite* (the relation between a pair may be a disjunction and not a primitive relation, as we have already seen) or even *inconsistent* (when the relation between two objects contradicts their relations with a third object). Figure 7 illustrates a multimedia document that perfectly matches the spatio-temporal structure of the query.

---

2 In [11], retrieval using MBR-based data structures is described, assuming 2D projection-based definitions of relations.

3 Text information retrieval techniques, deal with the problem of similarity by associating the retrieved documents with a score proportional to the similarity of the query and the document [13].

In general, multimedia queries can be formalised as tuples (*i*, *degree*, {(*X*, *Y*, *r*)}), where *i* is a set of documents, *degree* is the maximum tolerance and {(*X*, *Y*, *r*)} is a finite set of 3-tuples, where *X* and *Y* are multimedia object classes or instances and *r* is a relation set. Each pair (*X*,*Y*) must satisfy *r*. For example, the previous query, assuming a tolerance of 2 and a repository of 2 documents, D_1, D_2, can be formally expressed as:

$Q_I = (\{D_1, D_2\}, 2, \{(I, T, \{R_{11111\text{-}11111\text{-}01111}, R_{11111\text{-}11111\text{-}11111}\}),$
$(I, V, \{R_{11111\text{-}11111\text{-}11110}, R_{11111\text{-}11111\text{-}11111}\}), (T, V, \{R_{10000\text{-}U\text{-}11000}, R_{11000\text{-}U\text{-}11000}\})\})$

During the execution of the query, multimedia documents are sequentially examined and different instantiations of pairs of objects are assessed for matching

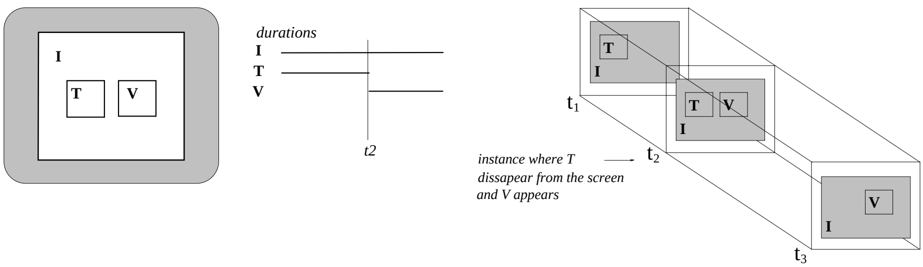


Fig. 7. A spatiotemporal query

each of the above tuples. Consider a generic set of query relations $Q=\{(X_i, Y_i, r_i) \mid i=1..n\}$ and a particular document instantiation $D = \{(X_{I\text{-}i}, Y_{I\text{-}i}, R_i) \mid i=1..n\}$. A potential similarity measure is:

$$S(D,Q) = \max_{\text{for all instantiations}} \left(\frac{\sum_{i=1}^{n} (Max\_Distance - d(R_i, r_i))}{n} \right)$$

where *Max_Distance* is the maximum distance in the conceptual neighborhood graph, which is 8 for Allen's 1D relations, 16 for their corresponding 2D extensions and 24 for the current application. *S*(*D*,*Q*) is equal to the similarity of the best instantiation. Its maximum value is equal to *Max_Distance* when all X_i can be instantiated so that all query relation sets are totally satisfied. Several other measures [6] can be defined accordingly using the neighborhood graph.

Applying the previous query and similarity measure to the document of Figure 8 (containing two text windows T_1, T_2, a video window *V* and two images I_1, I_2) we get two possible instantiations for the text window object and two instantiations for the image object, resulting in a total of four instantiations. Each instantiation produces a different subdocument with its own similarity.

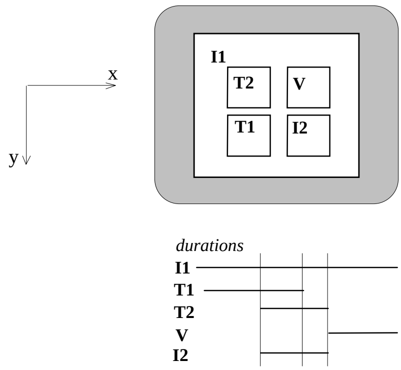
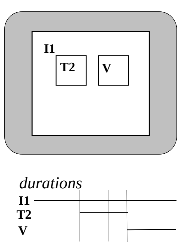


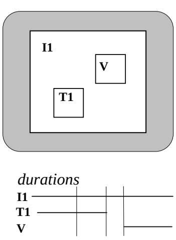
Fig. 8. Example multimedia document

The user may impose a degree of acceptance, so the result consists of all subdocuments that have a difference from the target score less than the given degree. In the example of Figure 9, if the given degree is 1, the first two subdocuments will be returned to the user.



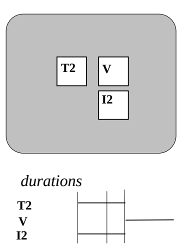
$R_{11111\text{-}11111\text{-}11111}(I_1,T_2)$, $R_{00100\text{-}00100\text{-}11110}(I_1,V)$, $R_{10000\text{-}01110\text{-}11000}(T_2,V)$

score = (24+24+24)/3 = 24



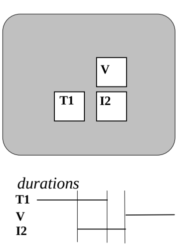
$R_{11111\text{-}11111\text{-}11111}(I_1,T_1)$, $R_{11111\text{-}11111\text{-}11110}(I_1,V)$, $R_{10000\text{-}00001\text{-}10000}(T_1,V)$

score = (24+24+23)/3 = 23.67



$R_{00001\text{-}00001\text{-}01110}(I_2,T_2)$, $R_{01110\text{-}00001\text{-}11000}(I_2,V)$, $R_{10000\text{-}01110\text{-}11000}(T_2,V)$

score = (15+16+24)/3 = 18.33



$R_{00001\text{-}01110\text{-}00111}(I_2,T_1)$, $R_{01110\text{-}00001\text{-}11000}(I_2,V)$, $R_{10000\text{-}00001\text{-}10000}(T_1,V)$

score = (17+16+23)/3 = 18.67

Fig. 9. Object Instantiations and Corresponding Scores

Structural queries involving classes rather than instances, are in general, computationally intractable because of the possible multiple instantiations. Let n_x be the number of objects of class C_x that appear in query Q, and N_x be the number of objects of C_x in document D. Then the matching of Q with D will have $N_x!/(N_x-n_x)!$ possible instantiations for the objects of type C_x (the number of n_x-permutations in a N_x-set). The total number of instantiations is:

$$\prod_{\text{Class } Cx \text{ appears in } Q} \frac{N_x!}{(N_x - n_x)!}$$

Although screen size restricts the number of participating multimedia objects, the exponential structure may be problematic for long-lasting documents. In general, multimedia similarity retrieval can be thought of as a constraint satisfaction problem (CSP) where the query objects correspond to variables and the document objects to their potential domains. The constraints are *soft*, meaning that potential solutions may partially or totally violate some constraints.

We have so far extensively experimented with 2D similarity problems, employing R-trees for spatial indexing and testing several constraint satisfaction algorithms such as dynamic backtracking, backjumping and forward checking (see [3]) modified for soft CSPs, and observed that most types of frequently used queries can be answered in real-time even for large document collections. Because of the structure of documents and queries, in most situations a large number of disqualify early so the search space is pruned very effectively. A thorough algorithmic description for spatial similarity and experimental results can be found in [9]. We are currently experimenting with higher dimensions and the incorporation of appropriate multimedia indexing techniques (see for example [15]) to guide search.

4 Conclusion

In this paper we propose a framework for processing multimedia similarity queries. A salient feature of such queries is the categorisation of results according to a variable degree of satisfaction. In our case, multimedia documents are assessed according to their spatiotemporal structure (the spatiotemporal relations among their constituent objects). We extend the concept of conceptual neighborhood (originally proposed for capturing similarity among temporal relations) in order to handle relations at various resolution levels, in multiple dimensions, and show how spatial and temporal inference can be significantly facilitated. Viewing multimedia documents as three-dimensional structures (collections of two-dimensional objects over time) we apply the above framework for effective processing of structural queries, i.e. queries of the form "find all documents containing a particular set of objects related through a specific spatial arrangement and temporal synchronisation". The usefulness of this approach can be stressed considering the recent large availability of multimedia documents in the WWW and the emerging need for navigation and retrieval based on such advanced information as structure, in addition to content.

At the implementation level we have mapped the problem of similarity to that of constraint satisfaction. Experimentation on several relevant algorithms has yielded promising results despite the exponential nature of the problem. Additionally, one of the most fruitful and interesting future research directions is the coupling of our techniques with appropriate query languages. At the lower level our model handles 1D primitive relations while at the user level one has to express spatial and temporal constraints in either a verbal query language or a pictorial one. The former [12] makes query formulation complicated and counter-intuitive since the relations among numerous variables (multimedia document objects) have to be explicitly expressed and predicates with undefined or ambiguous semantics (like *northeast, overlaps, follows, before, far*, etc.) have to be mapped to appropriate relation sets. Pictorial languages [14] are more friendly and help avoid inconsistencies but are restrictive, since sketching objects on a board implies exact relations (topological, approximate distance, etc.) which may not be always desirable (the user may not want to specify some exact relations). For multimedia queries matters are even more complicated since they should combine both spatial and temporal elements. One possible solution is the separation of the query's spatial and temporal aspects in two different query languages and appropriate selection of a suitable language for each of the two.

Acknowledgements

This work was done when the first author visited the Hong Kong University of Science and Technology, supported by DAG96/97.EG36 from Hong Kong Research Grant Council. The authors would like to thank Nikos Mamoulis for proof-reading the paper and providing useful comments.

References

1. Allen, J.: Maintaining Knowledge About Temporal Intervals. CACM, **26(11)** (1983)
2. Bruns, T.H., Egenhofer, M.J.: Similarity of Spatial Scenes. In Proceedings of the 7th Symposium on Spatial Data Handling (SDH) (1996)
3. Dechter, R.: Enhancement schemes for constraint processing: Backjumping, learning, and cutset decomposition. Artificial Intelligence **41** (1990) 273-312
4. Faloutsos, C., Equitz, W., Flickner, M., Niblack, W., Petkovic, D., Barber, R.: Efficient and Effective Querying by Image Content. Journal of Intelligent Information Systems **3** (1994) 1-28
5. Freksa, C.: Temporal Reasoning based on Semi Intervals. Artificial Intelligence **54** (1992) 199-227
6. Jain, R., Murthy, J., Tran, P., Chatterjee S.: Similarity Measures for Image Databases. FUZZ-IEEE (1995)
7. Megalou, E., Hadzilacos, Th., Mamoulis, N.: Conceptual Title Abstractions: Modeling and Querying Very Large Interactive Multimedia Repositories. 3rd International Conference on Multimedia Modelling (1996)
8. Nabil, M., Ngu, A., Shepherd, J.: Picture similarity retrieval using 2d projection interval representation. IEEE TKDE **8(4)** (1996)
9. Papadias, D., Mamoulis, N., Delis V.: Algorithms for Querying by Spatial Structure. International Conference on Very Large Data Bases (VLDB'98) (1998)

10. Papadias, D., Sellis, T.: Qualitative Representation of Spatial Knowledge in Two-Dimensional Space. Very Large Data Bases Journal **3(4)** (1994) 479-516
11. Papadias, D., Theodoridis, Y.: Spatial Relations, Minimum Bounding Rectangles, and Spatial Data Structures. International Journal of Geographic Information Systems **11(2)** (1997) 111-138
12. Roussopoulos, N., Faloutsos, C., Sellis, T.: An Efficient Pictorial Database System for PSQL. IEEE Transactions on Software Engineering **14(5)**, (1988) 639-650
13. Salton, G., Allan, J., Buckley, C.: Automatic Structuring and Retrieval of Large Text Files. Communications of the ACM **37(2)** (1994) 97-108
14. Smith, J. R., Chang, S-F.: Searching for images and videos on the World-Wide Web. Technical Report CU/CTR 459-96-25 Columbia University (1996)
15. Vazirgiannis, M., Thedoridis, Y., Sellis, T.: Spatiotemporal Composition and Indexing for Large Multimedia Applications. to appear in ACM/Springer Multimedia Journal (1998)