Similarity-based Retrieval of Temporal Specifications and its Application to the Retrieval of Multimedia Documents

K. Selçuk Candan* and Prakash Yamuna
Computer Science and Engineering Department,
Arizona State University
Tempe, AZ 85287, USA.
Email: {candan,prakash.yamuna}@asu.edu
Phone: +1 (480) 965-2770, Fax: +1 (480) 965-2751

Abstract

In this paper, we describe a similarity-based retrieval framework for temporal information, such as multimedia presentations. We develop techniques that allow users to query and retrieve multimedia documents, based on their temporal content. For this purpose, we describe different temporal data models and a set of similarity metrics applicable for different retrieval tasks. We develop algorithms that efficiently compute these metrics and report on experiment results. We also develop algorithms that efficiently index temporal structures based on these measures and show that the proposed variant of multi dimensional scaling is efficient and provides high quality retrieval of temporal specifications.

Keywords: Similarity-based retrieval, temporal information, multimedia retrieval

1 Introduction

Most current multimedia object and document management systems have major shortcomings. These systems are either single-media and single-feature based or they ignore the temporal and interaction structures of the documents they store during retrieval. Even those systems which consider the temporal structure of documents do not use temporal information for *similarity-based* retrieval task (although similarity-based retrieval is very common for others, such as retrieval based on spatial information).

In earlier work, we focussed on the authoring of flexible multimedia documents through a prioritized constraint-based temporal language and user interface [1, 2] and on querying and retrieval of media objects, such as images and videos [3, 4, 5]. In this paper, we develop techniques that allow users to query and retrieve temporal specifications. Below, we describe two motivating applications.

*Communicating author

1.1 Application I: Multimedia Databases

Multimedia systems employ similarity based retrieval techniques to retrieve images, video, and other media objects. We view a multimedia document as a collection of media objects, which are interlinked with each other through various structures including temporal, spatial, interaction, and user interaction structures. Therefore, in order to be able to retrieve multimedia documents the way we retrieve individual media objects, we need to develop similarity-based retrieval techniques that apply to these structures.

1.2 Application II: Schedule Databases

In order to improve their efficiency, many companies have to deal with scheduling problems. For example a manufacturing company with an automated manufacturing process will have scheduling problems dealing with its factory processes. A need for managing these schedules in an efficient manner is apparent. In fact in a recent work Adalı et al. [6] propose plan databases to manage plans. We believe that, since many scheduling tasks are potentially similar to each other, such a database should also support similarity based retrieval. For instance, given a task to schedule, a user may want to find already created schedules that he or she can use as a starting point that can be tailored to the new task. In such a scenario (1) a user may provide the system with a rough sketch of his problem, identifying critical aspects of the problem and when (2) the system returns schedules that are relevant, (3) the user may then select a schedule appropriate and modify it to suit the current needs. Another application is task management, where users maintain versions of a schedule and may need to identify how much each version differs from the rest.

The techniques introduced in this paper will enable advances in (a) multimedia document authoring and presentation, and (b) indexing, querying, and retrieval of database objects that contains temporal specifications. We describe distance measures that consider user's intentions and we develop algorithms that efficiently compute these measures.

2 Background and Related Work

A multimedia object and document management system must provide the following functionalities:

- *Authoring, storage, and rendering of distributed, interactive multimedia documents:* Authors must be able to specify the spatial and temporal contents of multimedia documents using a multitude of models. Multimedia documents created by this system must scale to the needs, resources, and preferences of the users.
- *Specification, processing, and refinement of object queries, and retrieval of media objects and documents:* The system must allow users to specify the contents of media objects/documents to be retrieved and be included in multimedia documents. Both media object and multimedia document retrieval tasks must be similarity-based. This will facilitate multimedia document retrieval based on, not only textual content, but also on media objects contained within documents and on temporal, spatial, and interactive document structures.

We see that a symbiosis between a *document data management system* and a *media-object database* will be beneficial for both of these systems: (1) a document management system is required by the document-base to process inputs and to help visualizing query results in the form of an interactive presentation that will guide users within the solution space. On the other hand, (2) a media object-base is required by the document management system as a provider

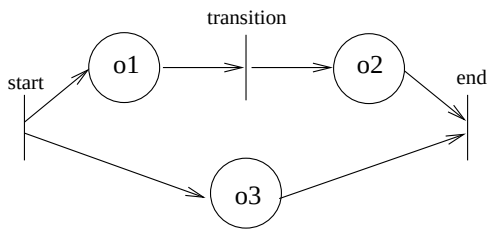


Figure 1: An interval-based OCPN graph

of media object and document content and as a storage means for cached objects and documents. Although such a system should also provide single media object retrieval services, in this paper we focus on the temporal multimedia document retrieval task.

2.1 Temporal Models

A multimedia document typically consists of a number of media objects that must be presented in a coherent, synchronized manner. Consequently, hierarchical, temporal, spatial, and user interaction structures, and QoS requirements have to be described and managed for multimedia documents.

Various markup languages, such as SGML and XML, provide suitable tools to facilitate multimedia document specifications. For instance, the HyTime standard [7] uses SGML to describe temporal properties of multimedia documents. A more recent standard proposed for multimedia document specification, SMIL [8, 9], uses a more restricted version of SGML, namely XML, for developing a general purpose media synchronization language. SMIL provides mechanisms to describe spatial, temporal, and interactive (in terms of hyperlinks) properties of multimedia documents.

There are various models that one can use to describe the temporal content of a multimedia document. The most basic model that addresses the temporal needs of multimedia applications is the *timeline model*. In this model, the user places events and actions on a timeline. Due to its simplicity, this model forms the basis for many academic and commercial multimedia authoring systems, such as the Athena Muse project [10], MacroMind Director [11]. Multimedia and Hypermedia information coding Expert Group (MHEG) defines a synchronization standard based on virtual timelines [12]. Similarly, Hytime standard [7], which is based on SGML [13], uses virtual timelines. Unfortunately, the timeline model is too inflexible for many applications. In [14], Hamakawa suggests an extension to the timeline model which uses *temporal glue* to allow individual objects to shrink or stretch as required.

A more flexible formalism is proposed by Allen [15, 16]. In this formalism Allen provides thirteen qualitative temporal relationships (such as *before*, *meets*, and *overlaps*) that can hold between two intervals. In [17], Little and Ghafoor propose an interval based model using object composition petri nets (OCPN, Figure 1), a modification of timed petri net model. In [18], Little and Ghafoor introduce another interval based conceptual model which can handle n-ary relationships among intervals and which can support user interactions. Prabhakaran and Raghavan also extend the OCPN model to handle similar user interactions [19]. Li, Karmouch, and Georganas, on the other hand, propose a Time Flow Graph (TFG) based model [20] which also is based on intervals. A recent multimedia synchronization standard, SMIL [8, 9], which is based on XML [21], use intervals and timelines to describe contents of multimedia documents. Interval algebra, however, is not perfect. In [22], Vilain and Kautz show that determining

the satisfiability of interval-based assertions is NP-Hard. Interval scripts [23], a methodology proposed to describe user interactions and sensor activities in an interactive system, benefits from a restriction on the allowed disjunction combinations in rendering the problem more manageable [24].

Several other researchers, including *ourselves* [1, 25], Buchanan and Zellweger [26, 27, 28], and Kim and Song [29, 30, 31], on the other hand, proposed the use of a highly-structured class of linear constraints called *difference constraints*, based on the instant-based point formalism proposed by Vilain and Kautz [22]. In [32], Peter van Beek provides a sound and complete algorithm for instant-based point algebra. Aspvall and Shilochi [33] and Dechter, Meiri, and Pearl [34] study algorithms for (temporal) constraint satisfaction problem and propose graph theoretical solutions. constraint-based approaches for document authoring and presentation includeÖzsoyoğlu *et.al* [35, 36, 37], and Vezirgiannis *et al*. [38]. Del Bimbo *et al.* [39] and others used temporal logic in retrieval of video data. More recently, [40, 41, 42, 43, 44, 45, 46] introduced alternative models, interfaces, and algebra for multimedia document authoring and synchronization.

In [1, 25, 47, 2, 48, 49, 50, 51, 52, 53, 54, 55], we presented various aspects of a *flexible* multimedia document authoring and presentation system, namely the Collaborative Heterogeneous Interactive Multimedia Platform (CHIMP). This system allowed a given set of specifications to be inconsistent for various reasons: (a) authors may make mistakes; (b) automatic quality adjustment or format conversion may lead into changes in the temporal properties of objects; (c) constraints introduced by user interactions may be inconsistent with the rest of the constraints; or (d) simply constraints themselves may be consistent, but the presentation may not be realizable due to resource limitations. If such inconsistencies exist, CHIMP would warn the user about the existence of the inconsistency, suggest ways to remove the inconsistency, or take immediate action and remove inconsistencies optimally. This flexibility was essential in many of the applications that CHIMP was designed for, including result visualization in multimedia databases and collaborative, distributed authoring of multimedia documents. Other contributions of CHIMP included

- dynamic verification and modification of documents in response to changes in resource availabilities [2, 53].
- generation of retrieval schedules and transmission of objects for distributed document presentations [2, 48, 49, 52]. We also recently developed techniques for cache management for interactive multimedia presentations and for creation and visualization of multimedia views [54, 55].

In order to achieve the required flexibility, CHIMP used an instant-based model and a constraint-based authoring strategy; more specifically, it used difference constraints. The restricted structure of the difference constraints allowed the development of certain graph theoretical algorithms that implement the needed flexibilities.

2.2 Similarity-based Retrieval

The knowledge about the dynamic properties of multimedia documents, not only enables users to query and retrieve them in a more complete way, but many essential functionalities, such as (1) object prefetching for interactive document visualization, (2) result summarization/visualization, and (3) query processing for document retrieval, depend on its

- efficiency in extracting and representing dynamic information,
- speed in comparing two documents using this information, and

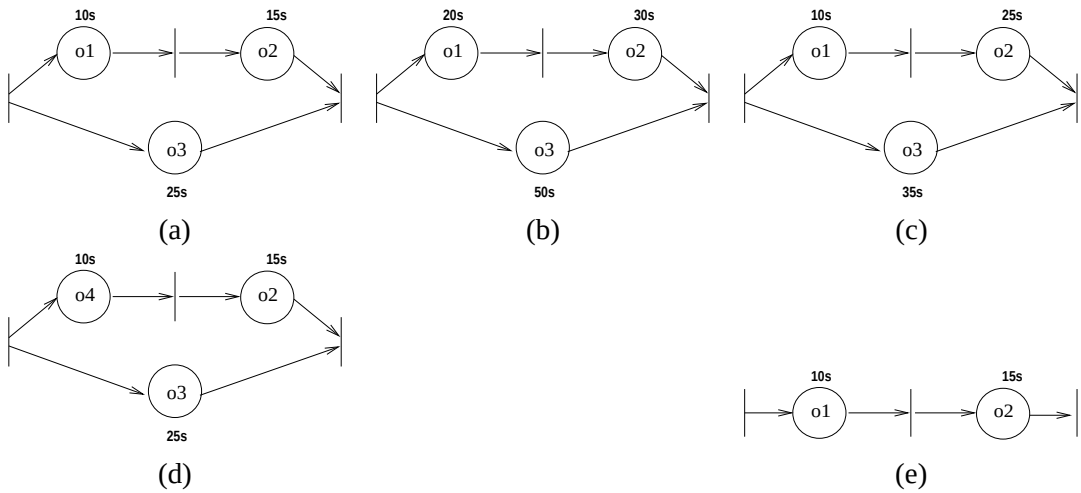


Figure 2: Five OCPN documents. Can we rank documents (b) to (e) according to their similarity to **(a)**? Hints: **(b)** has all object durations multiplied by 2, **(c)** has two objects with different, and one object with the same duration as (a), **(d)** has all object durations intact, but one of the object IDs is different, and **(e)** has a missing object.

- capability of providing a meaningful similarity value as a result of the comparison.

In the previous section, we have seen that the temporal characteristics of a document can be represented using *instant*-based models (e.g. difference constraint graphs) or *interval*-based models (e.g. OCPN). Although these two models show fundamental differences, for the purposes of this paper, we can abstract both of them as directed graphs, where the nodes are labeled with object names or descriptions and the edges labeled with numbers.

Definition 2.1 (Temporal Structure) Given a multimedia document d, its temporal structure can be represented as a directed graph $T_d(V, E, \rho, \mu, n_0)$, where V is the set of nodes, E is the set of edges, $\rho : V \rightarrow O$, is a mapping from nodes to object names/descriptors, $\mu : E \rightarrow R$ is a mapping from edges to real numbers, and $n_0 \in V$ is a start node. Let us denote the space of all such structures, $\mathcal{T}$. $\diamond$

Since document retrieval may require similarity-based comparison of various structures, a multimedia retrieval system must employ suitable comparison metrics. Figure 2 shows five OCPN documents to demonstrate the difficulty of defining similarity metrics for temporal multimedia documents. Note that the similarity (or the distance) between graphs have been studied by various researchers. Due to their restricted structures, trees and acyclic graphs have been the most studied types of graphs. In the most restricted form, Tai [57] and Shasha and Zhang [58, 59] provide postorder traversal-based algorithms for calculating the *editing distance* between ordered, node-labeled trees. The editing distance is defined by the number of *insert*, *delete*, and *modify* operations. In [60], Zhang, Wang, and Shasha extend their works to connected, undirected, acyclic, graphs where only edges are labeled. They first show that the problem is, as expected, NP-hard and then they provide an algorithm for computing the edit distance between graphs where each node has at most two neighbors. Chawhate *et al.* [61, 62] provide alternative, and more flexible, algorithms to calculate the edit distance between ordered node-labeled trees. Other research in tree similarity include [63, 64, 65, 66].

Most of these algorithms do not consider the edge labels, especially not the quantitative nature of them. Furthermore, these algorithms work for trees, not arbitrary graphs. Although the edit distance concept may still be useful

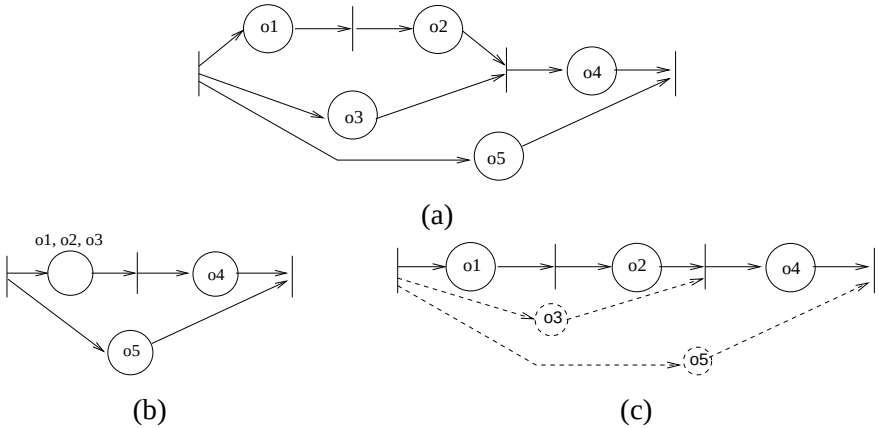


Figure 3: An OCPN document **(a)** and two possible summarizations. **(b)** when the temporal relationships between objects $o1$, $o2$, and $o3$ are not relevant; and **(c)** would be meaningful if objects $o3$ and $o5$ are not important.

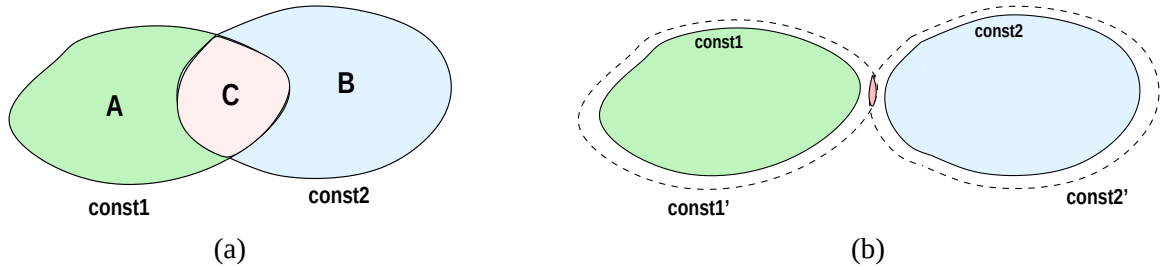
for some type of graphs, it needs to be extended to handle quantitative information. For example, the temporal graph in Figure 2(b) is structurally very similar to the one in Figure 2(a); except that the all temporal durations are twice as large in Figure 2(b). In order to account for these kind of similarities, we need novel algorithms which can handle temporal graphs.

Note also that the edit distance is not the only way to measure similarity between graphs. One other way to do so is to rewite a given temporal graph in terms of a conjunctive statement, where predicates correspond to the nodes and edges; and then to use fuzzy logic techniques, such as the ones we proposed in [67, 68], to evaluate the conjunct. The main differences of this approach from the edit distance measurement is that (a) well studied fuzzy query processing techniques can be used to process queries and (b) two graphs which are graphically entirely different yet temporally equivalent may be accepted logically similar. These two differences, however, may or may not be desirable: First of all, we can not be sure that the general purpose query processing techniques for fuzzy queries will be as efficient as special purpose graph-based query processing techniques for temporal graphs. Secondly, depending on the application, the structure of the temporal graph may be more important in retrieval, than the actual temporal values.

3 Defining Temporal Similarity

Similarity-based retrieval of multimedia documents is a challenging task due to the involvement of temporal structures of the documents. On the other hand, it is an essential part of a multimedia document management system, as many functionalities, such as query processing for document retrieval, result summarization/visualization, and object prefetching for interactive document visualization, require a meaningful similarity value to compare documents and an efficient mechanism to compute such similarity values.

In most non-trivial models, dynamic properties are declared as relationships between the media objects. Therefore, a general model must enable us to compare two documents based on the declared *intentions* of the document authors. As a declaration mechanism, constraints have been used by various researchers, including ourselves, to describe the temporal information [34, 1, 26, 29]. In general, we can define a temporal constraint satisfaction problem (TCSP) as follows:



(a) (b)

Figure 4: Intensional (a) similarit yand (b) dissimilarity

Definition 3.1 (TCSP) *Given a 4-tuple $\langle \mathcal{C}, \mathcal{I}, \mathcal{E}, \mathcal{P} \rangle$, such that*

- $\mathcal{C} = \{C_1, \ldots\}$ *is an infinite set of temporal constants,*
- $\mathcal{I} = \{I_1, \ldots, I_i\}$ *is a set of interval variables,*
- $\mathcal{E} = \{E_1, \ldots, E_e\}$ *is a set of event variables,*
- $\mathcal{P} = \{P_1, \ldots, P_p\}$ *is a set of predicates, where each P_i takes a set of intervals from $\mathcal{I}$, a set of events from $\mathcal{E}$, and a set of* constants *from C, and evaluates to* true *or* false*,*

we can define a temporal constraint satisfaction problem as a conjunctive normal formula (CNF) over these predicates, variables, and constants.

Example 3.1 Let $\mathcal{C} = \{Z^+\}, \mathcal{I} = \{int(o_1), int(o_2)\}, \mathcal{E} = \{pres_{st}, pres_{end}, st(o_1), st(o_2), end(o_1), end(o_2)\}$, the following is a temporal constraint satisfaction problem (since the predicates are intuitive enough for our purposes, we will not define them):

$$\begin{aligned} T \;=\; & (before(int(o_1), int(o_2))) \wedge \\ & (\leq (pres_{st}, st(o_1), 3) \vee \leq (pres_{start}, st(o_2), 20)) \wedge \\ & (= (pres_{end}, end(o_2)). \end{aligned}$$

Note that, in this example, the events in $\mathcal{E}$ and intervals in $\mathcal{I}$ are not independent; for instance, the beginning of the interval $int(o_1)$ corresponds to the event $st(o_1)$. We can account for these using additional special predicates.

3.1 Similarity Models

Given the above definition of declarative temporal content of multimedia documents, we can introduce various models of temporal similarity:

- *Intensional Similarity.* According to this general definition of TCSP, each predicate evaluates to either *true* or *false*; and a problem is satisfiable only if the corresponding formula (i.e., each disjunct in the formula) evaluates to *true*. If there are multiple assignments that satisfy the TCSP, then the problem has, not one, but a set of solutions (without a loss of generality, we can assume that that this set is always finite). In a sense, the *semantics* of the document is described by the set of presentation solutions that the corresponding constraints allow.

In the case of the timeline model, each solution set contains only one solution; whereas more flexible models may have multiple solutions among which the most suitable is chosen based on user preferences or resource requirements. For example, Figure 4(a) shows the solution sets of two documents, Doc_1 and Doc_2. Here, C is the set of solutions that satisfy both documents, whereas A and B are the sets of solutions that belong to only one of the documents. We can define the *intensional* similarity of the documents Doc_1 and Doc_2 as

$$similarity(Doc_1, Doc_2) = \frac{|C|}{|A| + |B| + |C|}.$$

- *Intensional* Dis*similarity.* The intensional semantics given above, however, has some shortcomings: if an inflexible model (such as the very popular timeline model) is used, then since there is only one solution for a given set of constraints, $\frac{|C|}{|A+B+C|}$ will either evaluate to 1 or 0; i.e., two documents will either match perfectly or will not match at all. It is clear that, such a definition is not useful for *similarity*-based retrieval. Furthermore, it is possible to have similar documents that do not have any common solutions, yet they may differ only in very subtle ways. A complementary notion of dissimilarity (depicted in Figure 4(b)) captures these cases more effectively:
  - Let us assume that two document Doc_1 and Doc_2 are consistent. Then, since there exists at least one common solution, these documents are similar to each other ($similarity = 1.0$).
  - If the solution spaces of these two documents are disjoint, then we can modify (edit) the constraints of these two documents until their solution sets overlap.

  We can define the *dissimilarity* between these two documents as the minimum increase required in the sizes of the solution sets for the documents to have a common solution:

  $$dissimilarity(Doc_1, Doc_2) = (|A'| + |B'|) - (|A| + |B|),$$

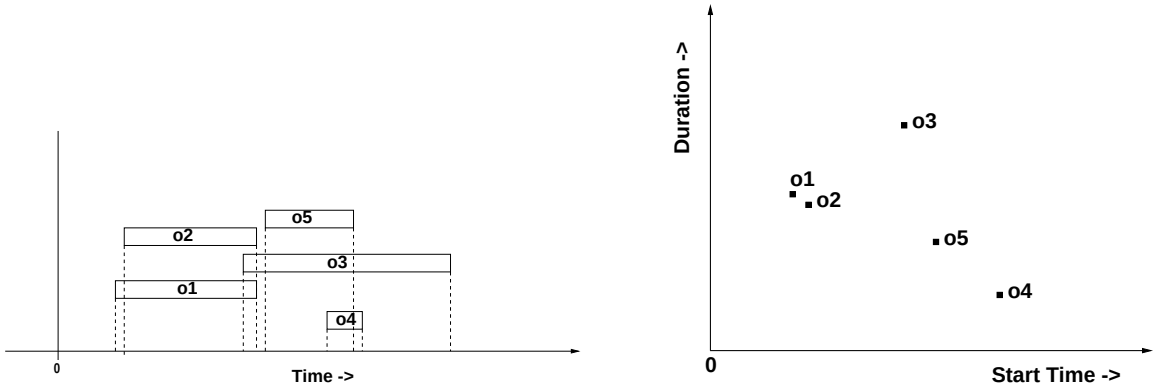
  where A' and B' are the new solution sets.

  The two intensional measures given above are complementary: one captures *the degree of similarity between consistent documents* an the other captures *the degree of dissimilarity between in-consistent documents*.

- *Syntactical Dissimilarity.* The two previous measures are (somewhat) related to the popular *edit distance* concept, which defines the dissimilarity between two strings as the number of changes necessary to make the two strings identical. The main difference, however, is that the two intentional measures are *semantical*, whereas the edit distance is purely *syntactical*: in the case of strings, edit distance can capture the syntactical relationship between strings `house` and `mouse`, but can not capture the semantical relationship between terms, *house* and *apartment*.

  Although the sequence of symbols (syntax) used gives only little clues about the semantics of the string, the set of constraints do actually *define* the semantics of the corresponding document. For instance, two seemingly different sets of temporal constraints can have the same solution set (hence the same schedule). Consequently, the dissimilarity of the constraints used by document authors does not represent the dissimilarity of the documents.

In the next subsection, we define the concept of document similarity for the simplest data model, namely the timeline model.



(a) (b)

Figure 5: (a) Specification of a multimedia document using the timeline model and (b) its representation in 2D space

3.2 Temporal Similarity - Timeline Model

The timeline model allows users to place objects on a timeline with respect to the starting time of the presentation. It is one of the simplest models and is also the least expressive.

3.2.1 Timeline Model

Figure 5(a) shows the temporal structure of a multimedia document according to the timeline model. The example document in this figure consists of six media objects with various start times and durations. Note that this representation assumes that no implicit relationships between objects are provided. Therefore, the temporal properties of the objects in a document can be represented as points in a 2D space, where one of the dimension denotes start time and the denotes the duration. Figure 5(b) shows the representation of the same document in a 2D space.

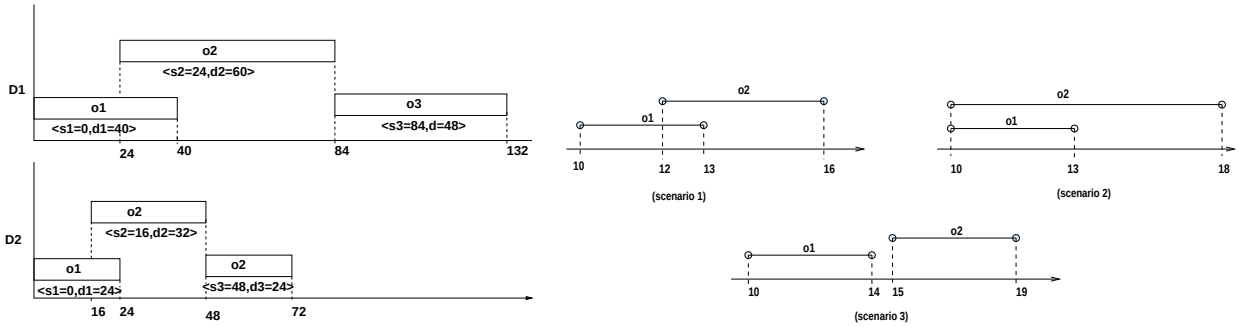
The temporal properties of a multimedia document is the combination of the temporal properties of the constituent multimedia objects and its duration. Each presentation object, o_i, within a document, D, can be defined as a 3-tuple of the form $\langle s_i^D, d_i^D, p_i^D \rangle$, where

- s_i^D denotes the presentation start time of the object,
- d_i^D denotes duration of the objects, and
- p_i^D denotes the priority/importance of the object in the presentation.

For the sake of simplicity, we will mostly use the simpler notations s_i, d_i, and p_i to denote the start time, duration, and priority of an object o. It should be noted however that all these are defined with respect to some multimedia document D. As a shorthand, we will use D also to represent the set of all presentation objects in the document. Furthermore, each document has a duration (Dp^D).

3.2.2 Distance Measures for Timeline Model

The first step in comparing two documents is to normalize (or scale) their durations.



(a) (b)

Figure 6: (a)Two documents that are equivalent after scaling and (b) three scenarios with equal city-block object distance, 5

Document Scaling. Temporal scaling is useful when users are interested in comparing the time in relative, instead of absolute, terms (Figure 6(a)). If the users would like the document similarity to be sensitive to the degree of scaling required, then we can define the scaling penalty, $\Upsilon(D_1, D_2)$, as the ratio of the document durations.

Object Mapping and Content-based Difference Between Objects. The next task is to identify which objects in the first document correspond to which objects in the second one. This task belongs to media-object database (MODB), which uses media features to compute most likely-to-be-related object pairs in two given documents. We can, then, define an object mapping based on the matching results returned by MODB. An object mapping, $\mu_{\{D1,D2\}}$, of documents D_1 and D_2, is defined as a set S of object pairs such that

- $\forall \langle o_i, o_j \rangle \in S, o_i \in D_1$ and $o_j \in D_2$ and $o_i, o_j \neq \phi$.
- $\forall \langle o_i, o_j \rangle, \langle o_k, o_l \rangle \in S, o_i = o_k \iff o_j = o_l$

It should be noted that the number of presentation objects in D_1 and D_2 may not be the same. Thus a mapping $\mu_{\{D_1,D_2\}}$ may or may not contain all the objects in D_1 or/and D_2. Furthermore, since media matching itself is similarity-based, given a mapped pair of objects, $\langle o_i, o_j \rangle \in S$, we can define a matching penalty $\lambda(o_1, o_j) \in [0.0, 1.0)$. This matching penalty is calculated using the feature-based similarity of the corresponding media objects calculated by MODB. Two objects that match perfectly have 0.0 penalty and two objects that do not match at all have 1.0 penalty.

Temporal Difference between Media Objects. Recall from Figure 5(b) that the temoral properties of presentation objects can be represented as points in a 2D space. Consequently, various distance measures, such as Minkowski distance ($[\ \mid s_i - s_j \mid^{\gamma} + \mid d_i - d_j \mid^{\gamma}\ ]^{\frac{1}{\gamma}}$), Euclidean distance ($[\ \mid s_i - s_j \mid^{2} + \mid d_i - d_j \mid^{2}\ ]^{\frac{1}{2}}$), or city block distance ($\Lambda(o_i, o_j) = [\ \mid s_i - s_j \mid + \mid d_i - d_j \mid\ ]$) can be used to compute the temporal difference ($\Lambda(o_i, o_j)$) between two objects o_i and o_j. However, the duration and the start times of a presentation object may have different importances in various applications (Figure 6(b)). Therefore, we can use weighted versions of these measures. Fore example, the *weighted city block distance* measure is defined as $\omega_s \times [\ \mid s_i - s_j \mid\ ] + \omega_d \times [\ \mid d_i - d_j \mid\ ]$.

Unmapped Objects. The object mapping, μ, may fail to map some objects that are in D_1 to objects in D_2 and vice versa. These unmapped objects must be taken into consideration when calculating the similarity/distance between two multimedia documents. In order to deal with unmapped objects, we will assume that each unmapped object,

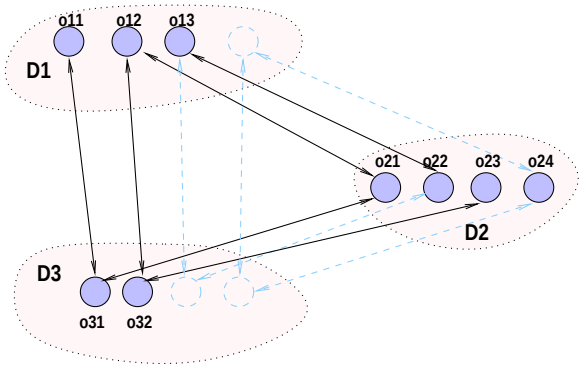


Figure 7: Three multimedia documents and the corresponding mappings. The dashed circles and lines show missing objects and the corresponding missing matchings.

$o_i = \langle s_i, d_i, p_i \rangle$, is mapped to a virtual object, $o_i^* = \langle s_i, 0, p_i \rangle$ in the other document. The values of $\Lambda(o_i, o_i^*)$ and $\Lambda(o_i^*, o_i)$ depend on the position of s_i and d_i. Figure 7 shows an example, where some objects in the documents are mapped to virtual objects in the others. Let us call the mapping from originally unmapped objects to virtual objects, $\mathcal{U}$.

Object Priorities and User Preferences. Given a mapping μ such that $(o_i, o_j) \in \mu$, we can calculate the priority, pri_{ij}, based on the priorities of both objects using arithmetic ($pri_{ij} = \frac{pr(o_i)+pr(o_j)}{2}$) or geometric ($pri_{ij} = \sqrt{(pr(o_i) * pr(o_j))}$) average of the object priorities.

Combined Timeline-based Distance Measure. We can define the distance ($\delta(o_i, o_j)$) between two objects as a combination of their temporal and the content-based distances. This distance can either be calculated by extending the dimensionality of the 2D temporal space with other features and applying a single distance measure to the combined space or by calculating the temporal and content-based distances separately and combining them afterwards:

$$\delta(o_i, o_j) = \omega_1 \times \lambda(o_i, o_j) + \omega_2 \times \Lambda(o_i, o_j), \tag{1}$$

where $\omega_1 + \omega_2 = \omega_1$. Then, we can define the distance ($\Delta(D_1, D_2)_\mu$) between two documents using the distances between mapped objects, penalties associated with unmapped objects, and scaling penalties:

$$\Delta(D_1, D_2)_\mu = \omega_3 \times \Upsilon(D_1, D_2) + \omega_4 \times \sum_{(o_i, o_j) \in \mu \cup \mathcal{U}} pri_{i,j} \times \delta(o_i, o_j) \tag{2}$$

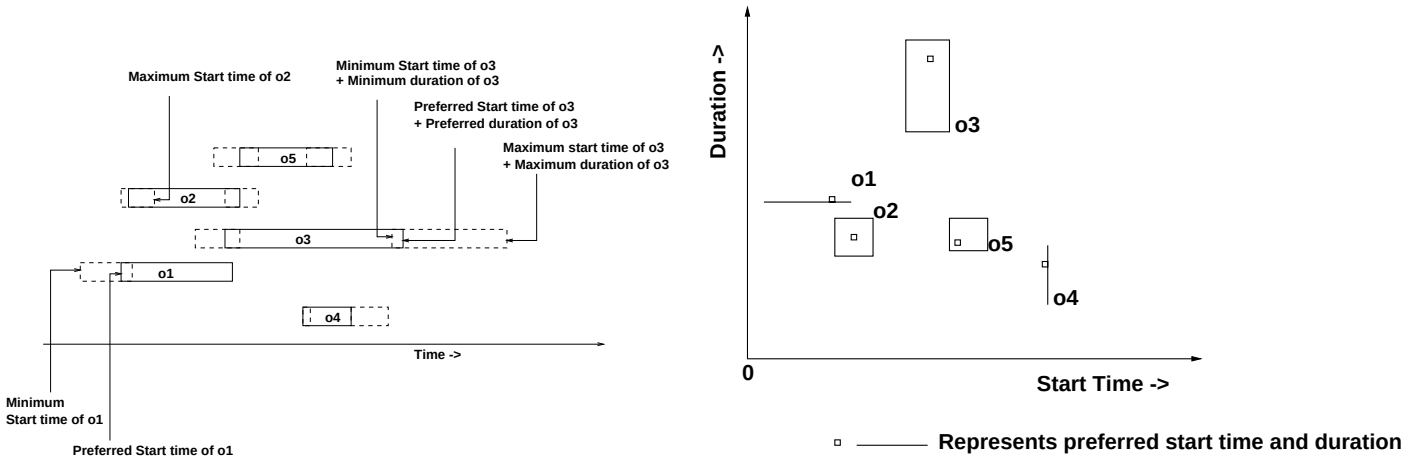
where $\omega_3 + \omega_4 = 1$.

3.2.3 Properties of the Distance Measure

The three desirable properties of a distance measure, Δ, are follows:

- $\Delta(x, x) = 0$,
- $\Delta(x, y) \geq 0$ for $(x \neq y)$, and
- $\Delta(x, y) = \Delta(y, x)$.

Furthermore, a fourth property (triangular inequality) is usually desirable for indexing purposes:



(a) (b)

Figure 8: (a) Representation of objects in extended timeline model and (b) 2D reprepresentation of the corresponding regions

- $\Delta(x, y) + \Delta(y, z) \geq \Delta(x, z)$.

This last property enables the use of spatial index structures, clustering algorithms, and nearest neighbor searches for similarity-based retrieval. The proposed distance very clearly satisfies the first three properties. The fourth property on the other hand may not always be satisfied.

3.3 Temporal Similarity - Extended (Flexible) Timeline Model

As mentioned in Section 3.2.1, the timeline is a rigid model. This means that the presentation of the object can not accomodate unexpected changes in the presentation specifications or in the available system resources. Consequently, various extensions to the timeline model have been proposed [14] to increase document flexibility. In this section, we use such an extention to show that increased flexibility in the timeline model gives rise to a distiction between what the authors intended and what the system provides; therefore complicating the definition of the distance measure.

3.3.1 Extended Timeline Model

As it was the case in the simple timeline model, each presentation object has an associated start time and a duration. However, instead of being scalar values, these paramaters are represented using ranges. This means that the presentation of an object can begin anytime during the valid range and the object can be presented for any duration within the corresponding range (Figure 8). Furthermore, each object also has a preferred start time and a preferred duration. Objects in a document, then, correspond to regions, instead of points, in a 2D temporal space. A flexible presentation object, o, can be defined as a triple of the form $\langle \mathcal{F}s_{\{s_{\lfloor}, s_{\perp}, s_{\rfloor}\}}, \mathcal{F}d_{\{d_{\lfloor}, d_{\perp}, d_{\rfloor}\}}, pr \rangle$, where

- $s_{\lfloor} \leq s_{\perp} \leq s_{\rfloor}$ – minimum, preferred, and maximum start times of o;
- $d_{\lfloor} \leq d_{\perp} \leq d_{\rfloor}$ – minimum, preferred, maximum durations of o;

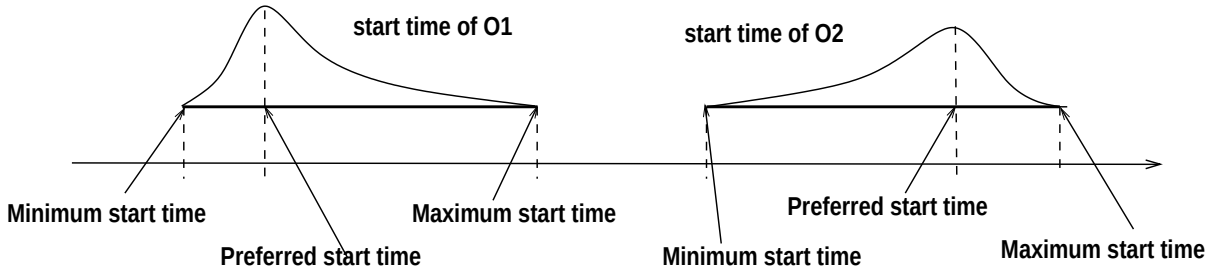


Figure 9: Start times of two flexible objects and the corresponding probability distributions

- $\mathcal{F}s_{\{s_\lfloor, s_\perp, s_\rfloor\}}$ is a probability density function such that
  - $\forall_{x<s_\lfloor} \mathcal{F}s_{\{s_\lfloor, s_\perp, s_\rfloor\}}(x) = 0.0$,
  - $\forall_{x<s_\rfloor} \mathcal{F}s_{\{s_\lfloor, s_\perp, s_\rfloor\}}(x) = 0.0$,
  - $\forall_x \mathcal{F}s_{\{s_\lfloor, s_\perp, s_\rfloor\}}(x) \leq \mathcal{F}s_{\{s_\lfloor, s_\perp, s_\rfloor\}}(s_\perp)$

  (for instance, it can have a normal distribution around the preferred time),
- $\mathcal{F}d_{\{d_\lfloor, d_\perp, d_\rfloor\}}$ is a probability density function with the same properties, and
- pr is the priority of o.

Figure 9 shows the start times of two example flexible objects. Intuitively, the probability density functions describe the likelyhood of the start time and the duration to take specific values. These functions return 0.0 beyond the minimum and maximum boundaries, and they return the maximum probability value for the preferred points. Note that even though the document authors usually does not specify a probability function, without the loss of generality, we can assume that such a distribution is *intended*, and infact exists due to system constraints.

3.3.2 Distance Measures for Extended Timeline Model

Similar to the simple timeline model, the main component of the document distance measure is the temporal distance between the mapped media objects. In order to calculate the temporal distance between two presentation objects, we can use the weighted city block distance (or any other suitable point distance measures). However, in this case, calculting the distance ($|s_i - s_j|$) between the start times or the distance ($|d_i - d_j|$) between durations is not straight forward. Since the start time of a flexible object can take any value between the corresponding $s_\lfloor$ and $s_\rfloor$, this fact has to be taken into consideration. Therefore, we define the distance between start times of two object o_i and o_j as

$$| \, s_i - s_j \, | \quad = \quad \int_{s_{i\lfloor}}^{s_{i\rfloor}} \int_{s_{j\lfloor}}^{s_{j\rfloor}} \mathcal{F}s_{i\{s_{i\lfloor}, s_{i\perp}, s_{i\rfloor}}(x) \times \mathcal{F}s_{j\{s_{j\lfloor}, s_{j\perp}, s_{j\rfloor}}(y) \times |x - y| \;\; dxdy. \tag{3}$$

Similarly, the distance between the durations of these objects can be defined using the duration boundaries and the duration probability functions.

One interesting consequence of this definition is that, although their description are identical, the start time of two objects, o_1 and o_2, shown in Figure 10 have a distance larger than 0. The reason for this is that although the descriptions of the start times are identical in terms of boundaries, preferred values, and probabilty distributions, there

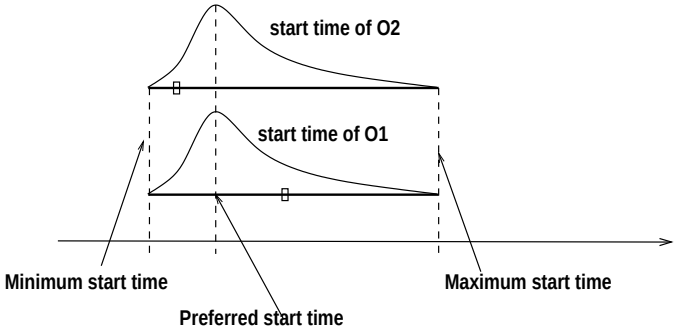


Figure 10: Start times of two identical flexible objects (note that the minimum, preferred, and maximum start times of both objects are identical); the two small rectangles on the other hand depict a possible scenario where the two objects start at different times.

is a chance that when presented to the user, these two object will not start at the same time. Hence, altough *intentionally* speaking, the distance between the start times should be 0, the observed difference is non-0. Consequently, when a flexible document is compared with itself, the document distance may be non-zero:

- $\Delta(D, D)_\mu \neq 0$.

This means that, although it is meaningful and intuitively attractive, an extensional (or observation-centric) definition of distance may not be suitable for retrieval of flexible documents. Instead, we need a model where intentional and extensional definitions of flexibilty are more integrated. In the next subsection, we will discuss such a model.

3.4 Temporal Similarity - Constraint-based Temporal Models

In Section 3, we have introduced the temporal constraint satisfaction problem (TCSP) as a 4-tuple $\langle \mathcal{C}, \mathcal{I}, \mathcal{E}, \mathcal{P} \rangle$ consisting of constants, interval variables, event variables, and predicates, respectively. TCSP can be used as a general framework for describing temporal content of multimedia documents.

3.4.1 Result Quality and its Relation to Similarity

In order to introduce further flexibility, we can relax the requirement that predicates return only *true* and *false*; and, instead, we can have predicates that will evaluate to a value within the interval $[0, 1]$, where 1 corresponds to *true* and 0 corresponds to *false*. In other words, we can say that each predicate may be partially satisfied and we can use fuzzy algebra to identify whether a temporal constraint satisfaction problem is satisfiable beyond a certain acceptability threshold. Another way to relax this definition is to keep the boolean nature of the predicates, but allow a certain number of the subcondtitions evaluate to false. Such a parial relaxation would allow the system to tackle the problem when it may not be possible to satisfy the original problem at its entirety. Clearly, the relaxed problem is of a lower quality than the original problem. Therefore, given a non-satisfiable constraint problem, it may be desirable to find a high-quality relaxation of it.

The notion of quality as described above is related to the notion of similarity. While an exact answer to a problem would be of high quality (satisfying all disjuncts), merely-similar documents (which do not satisfy all disjuncts) would be of lower quality.

Given two documents, D_1 and D_2, and corresponding temporal constraints, $C(D_1)$ and $C(D_2)$, and a mapping μ between the objects in these two documents, we can create a merged document, $D_{(1,2)}$, as follows:

- If (o_i, o_j) is in μ (note that $o_i \in D_1$ and $o_j \in D_2$), then
  - create a new object $o_{(i,j)}$ and insert into $D_{(1,2)}$
- If o_i is an unmapped object in D_1, then
  - create a new object $o_{(i,*)}$ and insert into $D_{(1,2)}$
  - create constraints $c_a = st(o_{(i,*)}) - et(o_{(i,*)}) \leq 0$ and $c_b = et(o_{(i,*)}) - st(o_{(i,*)}) \leq 0$.
  - insert c_a and c_b into $C(D_{(1,2)})$
- If o_i is an unmapped object in D_2, then
  - create a new object $o_{(*,i)}$ and insert into $D_{(1,2)}$
  - create constraints $c_a = st(o_{(i,*)}) - et(o_{(i,*)}) \leq 0$ and $c_b = et(o_{(i,*)}) - st(o_{(i,*)}) \leq 0$.
  - insert c_a and c_b into $C(D_{(1,2)})$.
- For every constraint, c, in D_1,
  - using the corresponding new objects in $D_{(1,2)}$ create an equivalent constraint, c', and
  - insert c' into $C(D_{(1,2)})$.
- For every constraint, c, in D_2,
  - using the corresponding new objects in $D_{(1,2)}$, create an equivalent constraint, c', and
  - insert c' into $C(D_{(1,2)})$.

Figure 11: Document merging algorithm (st denotes the start-time and et denotes the end-time of an object)

3.4.2 Calculating Temporal Document Distance based on Constraints

In this section, we describe how to calculate temporal document distances using the constraint-based representation of temporal documents [69]. Given a document, D, the corresponding set of constraints, $C(D)$, describe the *intension* of its author. Therefore, if two documents, $D1$ and $D2$ are identical or if they represent non-conflicting intensions of their authors, then when the two sets, $C(D_1)$ and $C(D_2)$, of constraints describing these intentions are put together (Figure 11), the resulting set of constraints should not contain any conflicts; i.e., the merged set of constraints should be satisfiable.

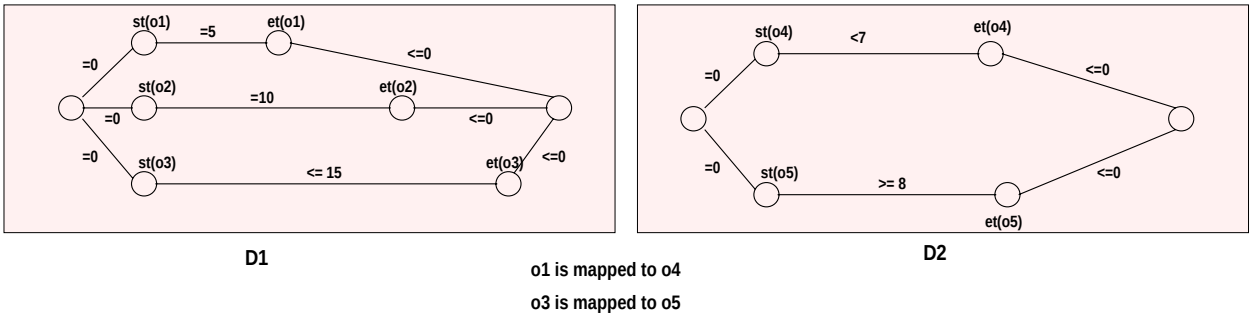
The document merging process presented in Figure 11 identifies those objects that are mapped to each other (i.e., that are identified to be identical) and merges the constraints in both documents corresponding to such mapped objects. For any unmapped object, it assumes that it is mapped to a 0 length object in the other document and creates an additional set of constraints that describes this fact.

Figure 12 shows two documents, an object mapping, and the corresponding merged document. Note that in this example, the only conflict is caused by the existence of the unmapped object: the constraints attached the object $o_{(2,*)}$ in the merged document are clearly inconsistent. This observation gives rise to a simple definition for a distance formula:

Simple Document Distance. The simple distance between two documents, D_1 and D_2, is defined as

$$\Delta_s(D_1, D_2)_\mu = total\_number\_of\_conflicts\_in(C(D_{(1,2)})). \tag{4}$$

This definition of the document distance is semantically very simple and clean. Furthermore, it satisfies the first



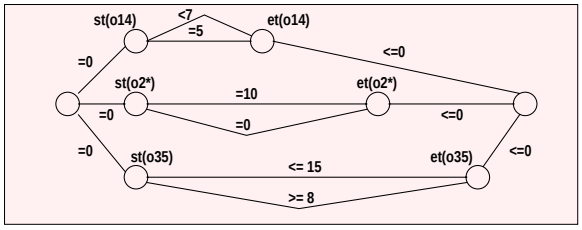
D1

D2

o1 is mapped to o4

o3 is mapped to o5

(a)



D(1,2)

(b)

Figure 12: (a) Two documents and (b) the corresponding merged graph (in order to minimize the number of edges in the graph, we did not use difference constraints in this figure; but, there are equivalents difference constraint graphs (st denotes the start-time and et denotes the end-time of an object)

three properties of distance measures:

- $\Delta(D_1, D_1) = 0$: when you merge a document with itself, you do not get any conflicts.
- $\Delta(D_1, D_2) \geq 0$ for ($D_1 \neq D_2$): when you merge two documents, the total number of conflicts is guaranteed to be 0 or a positive integer.
- $\Delta(D_1, D_2) = \Delta(D_2, D_1)$: independent of the order of the two documents, the document merging process will result in the same graph; therefore, the number of conflicts in both cases will be the same.

The triangular inequality, however, does not necessarily hold:

- $\Delta(D_1, D_2) + \Delta(D_2, D_3) \geq \Delta(D_1, D_3)$ **is false**: consider the following three constraints corresponding to three different documents containing the same object (st denotes the start-time and et denotes the end-time of an object):
  - $C(D_1) = \{et(o) - st(o) \leq 8\}$,
  - $C(D_2) = \{et(o) - st(o) \geq 5\}$, and
  - $C(D_3) = \{et(o) - st(o) \geq 10\}$.

  When combined, documents pairs D_1-D_2, and D_2-D_3 are conflict free. That is $\Delta(D_1, D_2) = \Delta(D_2, D_3) = 0$. However, when we combine D_1 and D_3 we get a set of constraints with a conflict; i.e., $\Delta(D_1, D_3) = 1$. Therefore, this definition of distance does not satisfy triangular inequality.

This result implies that we can not simply use spatial index structures or nearest neighbor searches for retrieval of temporal documents. The real disadvantage of this measure, however, is that it is very expensive to compute. It can be shown [2] that in the worst case, the number of conflicts in a document is exponential to the size of the document (in terms of objects and constraints). Therefore, this definition is not very practical. Therefore, we need an alternative measure which is easier to compute. In [2], we showed that under certain conditions it is easier to find an optimal set of constraints to be relaxed than to identify the total number of conflicts in the constraints. Therefore, it is possible to identify a set of edges/constraints whose removal would eliminate all conflicts. The minimum number of constraints that needs to be removed to achieve consistency, on the other hand, provides an indication about the reasons of conflicts.

Document Distance. The distance between two documents, D_1 and D_2, is defined as

$$\Delta(D_1, D_2)_\mu = min\_num\_of\_constraints\_removed(C(D_{(1,2)})). \tag{5}$$

Note that it can be shown that this distance measure also satisfies the first three properties of distance measures, yet it also fails to satisfy the triangular inequality.

In [2], in order to resolve conflicts during multimedia document presentations, we defined what it means to *optimally* remove the inconsistencies:

- a *card-optimal relaxation* of C is any subset $C' \subseteq C$ such that C' is solvable, and there is no other relaxation C'' such that $card(C'') > card(C')$. Here $card(C)$ is the number of constraints in C.
- a *priority-sum-optimal relaxation* of a set of constraints C is a consistent subset C' of C such that there is no other consistent subset C'' which satisfies $(\sum_{c \in C''} pri(c)) > (\sum_{c \in C'} pri(c))$ where $pri(c)$ denotes the priority of constraint c.
- a *priority-optimal relaxation* of a set of constraints C is a consistent subset C' of C such that if c is in $C - C'$, then there exists a set of conflicting constraints S in C where for all c_i in S, $pri(c_i) \geq pri(c)$.

The card optimal relaxation corresponds to the document distance definition we stated earlier. It is clear that, in order to deal with priorities, we should use the other priority-based relaxation definitions. Therefore, using these definitions, we can define a generic distance measure that captures different relaxation alternatives.

Θ-Document Distance. Given a condition (such as priority-optimality) Θ, the Θ-distance between two documents, D_1 and D_2, is defined as

$$\Delta_\Theta(D_1, D_2)_\mu = \Theta\_relaxation\_measure(C(D_{(1,2)})), \tag{6}$$

The significance of a removed constraint may be different depending on the size of the documents involved in the comparison. For a document with hundred constraints, removal of one constraint may not be significant; yet, for a document with with only a few constraints, removal of a single constraint may be relatively important. Therefore, we need to define a relative document distance.

Relative Θ-Document Distance. Given a condition (such as priority-optimality) Θ, the relative Θ-distance between two documents, D_1 and D_2, is defined as

$$\Delta_\Theta(D_1, D_2)_\mu = \frac{\Theta\_relaxation\_measure(C(D_{(1,2)}))}{non\_relaxed\_measure(C(D_{(1,2)}))}, \tag{7}$$

For instance, if we are using the card-optimal relaxation to define the distance measure, the denominator is *the number of constraints removed* and the nominator is *the number of constraints in the merged document*. $\Delta_{\ominus}(D_1, D_2)_{\mu}$ will always be between 0.0 and 1.0.

3.5 Summary

In this section, we developed a set of constraint-based distance measures that take into account the known relationships (or *intensions*) between presentation objects. We have shown that other temporal data models can be represented in terms of constraints; therefore enabling inter-model comparison of multimedia documents. Note that, in this section, we did not describe how this similarity measure can be used to index/cluster documents. Since none of the measures satisfy triangular inequality, spatial index structures can not be used. However, techniques, such as *multidimensional scaling*, can be adapted for retrieval.

4 Indexing and Clustering of Temporal Specifications

Most database systems employ index and clustering techniques to facilitate faster searching. This is due to the fact that database systems contain huge amounts of data and it is time consuming to perform a linear scan of the database before retrieving the relavant data. Since searching the entire database is not an acceptable solution, the idea is reduce the search space in such a manner that we move towards the relevant documents in the space and away from the irrelevant documents. In order to do so we need to order the documents based on some criteria if possible. In general there are two approaches to support similarity based retrieval.

- using implicitly defined distance measures, as in most graph-based clustering algorithms
- using features, as in vector-space retrieval approaches

In clustering based solutions, *similar* documents are grouped together forming a cluster. The idea behind clustering is that if there is a way to measure similarities (or distances) between pairs of objects, then we can group the objects into cluster and select a representative object, called *pivot*, for each cluster. Then, given a query, we can compare the query object to the *pivot* of each cluster, significantly reducing search time (Figure 13). Although clustering, especially when there are no explicit features of the objects to facilitate creation of index structures, is a viable option, it suffers from being rather static in terms of determining *cluster boundaries* and conservative in terms of pruning the search spaces.

The second approach is used whenever the data objects do not lend themselves to the extraction of features. Object indexing-based retrieval systems use characteristic features of the objects to place them into a search structure. Recently, some researchers have proposed converting the distances into points in a k-dimensional space, where k does not represent the number of explicit features of the data type, but the number of *characteristic* number of features for the given data set. In [71] Faloutsos et al. discuss about the problems associated with identifying features and more importantly in designing feature extraction functions. In the same paper,they propose a way to map objects into points in a k-dimensional space based on just the distance/similarity values between objects. They reason that it is far easier for domain experts to assess the similarity/distance of two objects than it is to identify features and design feature extraction functions. In addition to the difficulty in designing feature extraction

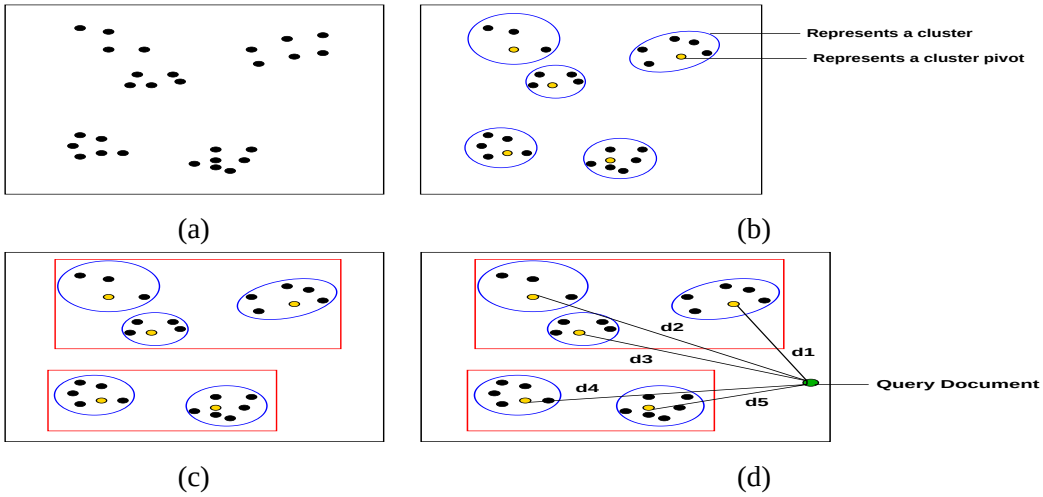


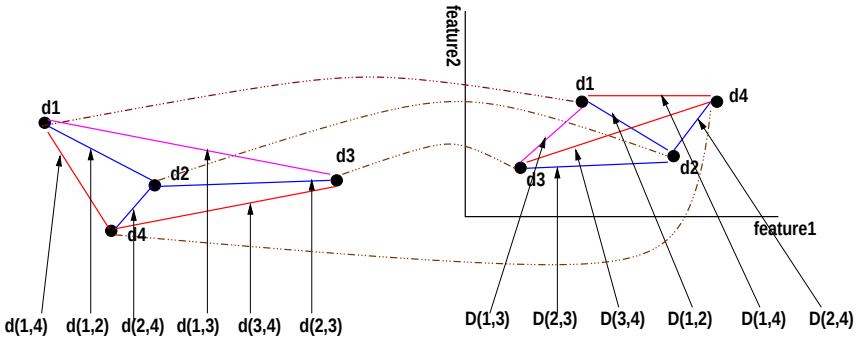
Figure 13: **(a)** A set of documents in a database, **(b)** the documents grouped together based on a "similarity" criterion, **(c)** a hierarchical arrangment of the clusters, and **(d)** a query document and distances to the various cluster pivots

functions, the index structures that have been developed for feature space can be directly applied on this unknown feature space. Their method is conceptually similar to the *Multi-Dimensional Scaling* approach [72, 73], however they provide a much more efficient way of mapping the objects into points in space. In this paper, we base our work on *Multi-Dimensional Scaling* approach as in [71], authors assume that the distance/similarity measure they have satisfies triangle inequality. An approach to handle the problem of triangle inequality has been to use modified distance measures that satisfy the triangle inequality. In [74] Fagin and Stockmeyer propose a way of handling triangle inequality. They propose a modified triangle inequality measure of the form $\Delta(O_1, O_3) \leq c(\Delta(O_1, O_2) + \Delta(O_2, O_3))$, where c is a constant that is not too large. Other approaches have been to add a constant to the triangle inequality equation. Given this modified distance function, clique based clustering algorithms could be applied as they provide much tighter bound clusters or method introduced in [71] to represent them in feature space can be applied.

4.1 Indexing Temporal Specifications

While approaches proposed in [74] are appropriate where they hold, we note that there may be many distance measures that will not satisfy the relaxed triangle inequality or even if they do the resulting distance measure may not be meanigful. A good example of a distance measure for which no form of relaxed triangle inequality holds good due to the very nature of its definition is the use of the number inconsistencies in a temporal constraint satisfaction problem that we proposed in Section 3.4.

While we could use graph-theoretical clustering algorithms, we need to consider the fact that most efficient index structures are space based. Thus instead of using graph clustering algorithms, we propose mapping temporal specifications into a k-dimensional space and using spatial index structures to facilitate efficient retrieval. For this purpose, we propose using an appropriately modified version of MDS [72, 73] to index temporal specificarions.



(a)

$$d(1,2) \approx D(1,2) \qquad d(1,3) \approx D(1,3) \qquad d(1,4) \approx D(1,4)$$
$$d(2,3) \approx D(2,3) \qquad d(2,4) \approx D(2,4) \qquad d(3,4) \approx D(3,4)$$

(b)

Figure 14: **(a)** Distance to feature space mapping, an initial configuration, **(b)** Intuition of MDS.

4.2 Standard Multi-Dimensional Scaling

Multi-Dimensional scaling (MDS) is an established concept introduced in the early 1960's [72, 73]. MDS is used for discovering the underlying spatial structure of a set of data items from the (dis) similarity information among them. There are many variations of MDS proposed in the literature [75, 76]. Two major types of MDS algorithms are *metric* and *non-metric MDS*. In metric MDS [75] the distance measures between objects are actual distance values, whereas in non-metric MDS [72, 73], distances are qualitative in nature; i.e., they are more of a ranking among rather than the actual distance between the objects.

Mapping from Distances to Space. MDS works as follows, it expects as inputs (1) a set of N objects, (2) a matrix of N x N, containing pair wise (dis)similarities, and (3) the desired dimensionality k. Given these inputs, MDS tries to map each object into a point in k-d space. The mapping process of documents, given their distances with respect to one another, is shown in Figure 14. The criteria for the mapping is to minimize a *stress* value which is calculated as

$$stress = \sqrt{\frac{\sum_{i,j}(d'_{ij} - d_{ij})^2}{\sum_{i,j} d_{ij}^2}},$$

where d_{ij} is the actual distance between objects O_i and O_j and d'_{ij} is the distance between points p_i and p_j. p_i represents objects O_i and p_j represents O_j in k-d space. Thus if we can maintain the distance between p_i and p_j the same as O_i and O_j then it it means we have $d'_{ij} = d_{ij}$. Therefore in this case the stress is 0.

MDS starts with an *initial configuration* of points. Many approaches have been proposed for identifying the *initial configuration*, the effect of the initial configuration is in limiting the number of iterations needed to achieve minimum stress. For the purpose of illustration we assume MDS uses a random configuration. Thus, MDS initially starts with a random configuration of points. It then applies the method of steepest descent iteratively to minimize

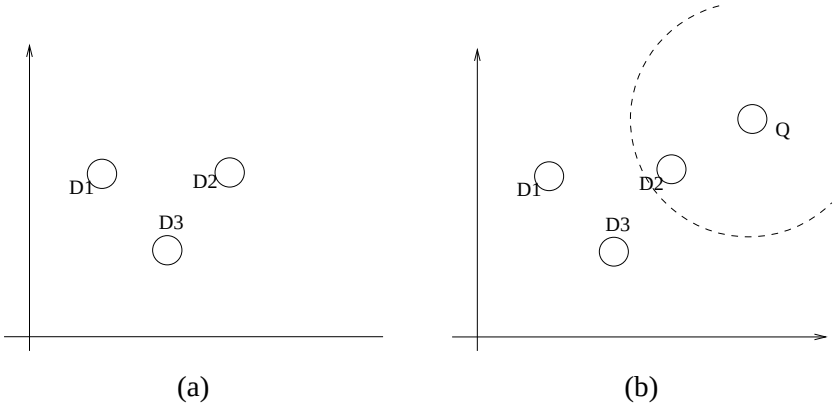


Figure 15: **(a)** Mapping of the three specifications in the Example 4.1 (D_1 and D_3 are the pivots) and **(b)** mapping of the query and the corresponding range search.

the stress. Note that, once the inputs are mapped into a k-dimensional space, one can use a multi-dimensional index structure to do nearest-neighbor and range searches.

Example 4.1 (MDS and Retrieval of Temporal Specifications) Let us consider the following three temporal specifications:

$D_1 : t_1 - t_2 < 5; \quad t_1 - t_3 < 3; \quad t1 - t_4 > 3$
$D_2 : t_1 - t_2 < 5; \quad t_1 - t_3 < 2; \quad t1 - t_4 > 1$
$D_3 : t_1 - t_2 < 5; \quad t_1 - t_3 > 5; \quad t1 - t_4 < 2$

If we use the $min\_num\_of\_constraints\_removed$ distance measure we have introduced in Section 3.4.2 to evaluate the temporal distance between these specifications, we have

$$\Delta(D_1, D_2) = 1; \Delta(D_1, D_3) = 1; \Delta(D_1, D_3) = 1;$$

that is, each pair of temporal specifications can be made consistent with eachother by removing one constraint. For example $t_1 - t_3 < 3$ in D_1 and $t_1 - t_3 > 5$ in D_3 are conflicting and removing any one of these will make the two temporal specifications consistent.

Let us assume that, based on these distance values, these three document are mapped into a 2-dimensional space by MDS as shown in Figure 15(a). Let us further assume that documents D_1 and D_3 are selected as the pivots. Now let us consider the following query specification:

$Q : t_1 - t_2 > 10; \quad t_1 - t_3 < 4; \quad t1 - t_4 < 3$

In this case, we have $\Delta(D_1, Q) = 2$ and $\Delta(D_3, Q) = 2$. Let us assume that based on these two disnace values, MDS maps temporal query Q into the 2-dimensional space as shown in Figure 15(b). If the search range is given as 1.5 units, the only temporal specification that will be returned is D_2. Note that, in this case, $\Delta(D_2, Q) = 1$ and D_2 is indeed a matching temporal specification. $\diamond$

Short-comings of MDS. While MDS does find the k-dimensional space for the temporal specifications provided as input, from the point of indexing, MDS suffers from two drawbacks:

- It requires $O(N^2)$ time to find the configuration of N points in a k-dimensional space.

- *Process* the given N temporal specifications and construct the $N \times N$ distance matrix required as input to MDS.
- Find the configuration (point representation of each document in a k-dimensional space).
- Identify c pivot/representative points (data elements), where each pivot p_i represent r_i many points.
- When a query specification q is provided, map the query into the MDS space using the c pivot points (accounting for r_i for each p_i). Thus the complexity of applying MDS is $O(c)$ instead of $O(N)$.
- Once the query is mapped into the k-dimensional place, use the spatial index structure to perform a range search in this space.

Figure 16: Extended MDS algorithm

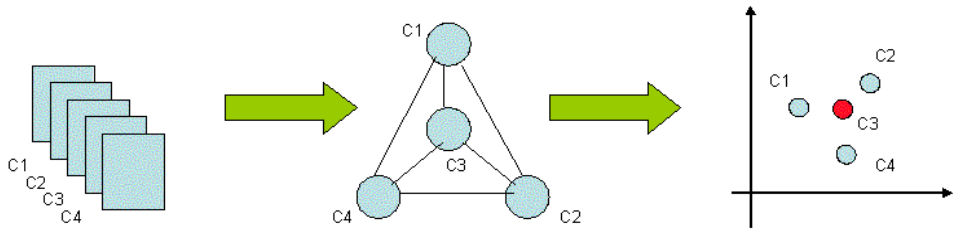


Figure 17: Temporal document indexing using MDS and pivot selection

- More importantly given a query, q, it takes $O(N)$ to map q into a point in k-dimensional space.

To understand why it takes $O(N)$ to find the spatial representation of q, note that the process that drives MDS is the distance between the documents. Thus given the configuration of points, we need the distance between q and all the objects in the database (N in this case), for MDS to be able to determine the spatial representation of q. Note that once we are able to map the N objects in the database and the query, q, in the same space k-dimensional space, we can use spatial index structures on this space to answer similarity-based retrieval queries. In the next section, we explain how we overcome some of the problems mentioned in this section for efficient similarity-based retrieval of temporal structures.

4.3 Efficient Multi-Dimensional Scaling

As we mentioned in 4.2, MDS suffers from two drawbacks: expensive (1) data-to-space and (2) query-to-space mappings. While the first drawback may be acceptable, since documents can be preprocessed as is done in many image databases where images are preprocessed, the real disadvantage is that to introduce the query object q into the k-dimensional space requires $O(N)$ time with a large constant. This implies that MDS is as bad as sequential scan.

In Figure 16, we propose an improved MDS algorithm [77]. The algorithm works by first mapping the data objects into a multidimensional space through MDS and selecting a set of objects as the pivots (Figure 17. In this particular case of multimedia documents with temporal characteristics, the first step of the algorithm involves

- finding the distances between the underlying temporal specifications of the documents using one of the algorithms described in the previous section, and
- creating the corresponding distance matrix using these distance values.

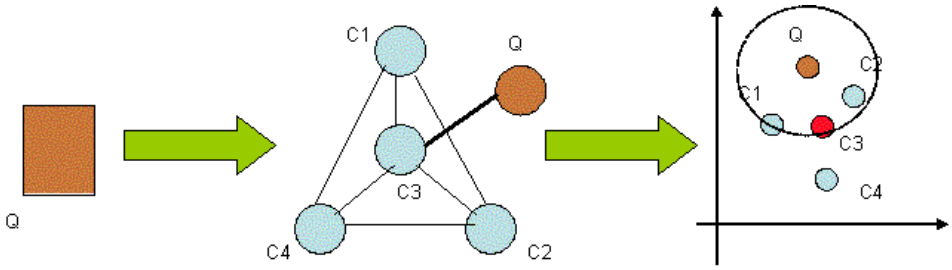
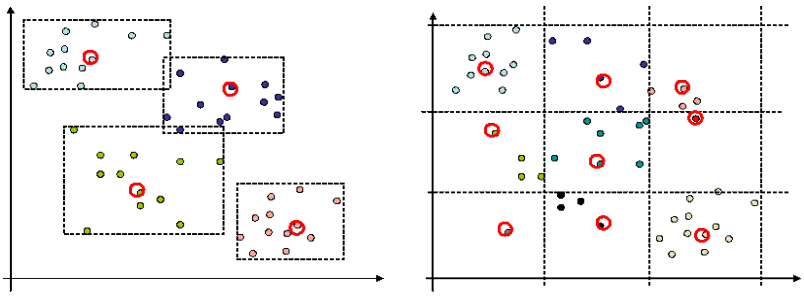


Figure 18: Mapping of the temporal query into the space using the pivot selected in the earlier phase selection



(a) (b)

Figure 19: **(a)**Data driven vs. **(b)** space driven decomposition

The temporal characteristics of the query object, then, is compared to the pivots and mapped into the same space as the other documents. (Figure 18). The query mapping is less accurate then the original data mapping as only the pivots are used for the mapping instead of the entire data set.

The advantage of the proposed method is that the complexity is reduced to the number of pivot points. Since this number can be very small compared to the actual number of documents in the database we end up with a sizable reduction in the query mapping complexity. However this approach clearly comes with a cost. Since we are finding the point representation of the query q in the space with respect to c points rather than N data points, the *tightness* for the point is comparatively low. Note that tightness refers to the set of points in the space that q can be mapped to. Greater the number of comparison points, greater the number of constraints on the number of points that q can be represented by and hence greater the possibility of a more accurate representation. However we note that if clusters are *good*, then they are a fair representation of the individual points they represent and hence the lesser the possibility of false drops or hits.

Note that, in this scheme, the quality of the retrieval will depend heavily on the c data points selected for the query-to-space mapping process. We see that there are two main approaches for selecting these points: **(1)** Data-driven and **(2)** space-driven (Figure 19).

Data-driven approach: We choose the c pivot points based on the distribution of the data elements, either using the original distances or using the MDS space. More specifically, in this approach, documents in the database are clustered, and cluster representatives are used as pivot points. The intuition is that the cluster driven selection of the points will have a better coverage of the data points. For this purpose, one option is to use clusters (minimum bounding regions) identified by a hierarchical multi-dimensional index structures (like the R-tree) that we use for

storing the MDS data points. For each cluster, we can use the data point closest to the center of the minimum bounding region as the cluster representative, and we can use these cluster representatives for the query-to-space mapping. The algorithm can be described as follows:

- Choose k nodes from R-tree such that all data points are covered (and each node corresponds to the same number of data points)
- For each cluster (R-tree node), compute its center
- Search for the leaf node in the Rtree closest to the center within the cluster and mark it as pivot

Space-driven approach: We subdivide the space and choose one data point to represent each space subdivision. The intuition is that the space driven selection of the points will have a better coverage of the space itself. We divide the space into uniform regions and for a randomly choosen k (k is less then equal to the number of uniform regions) region, we select a point closer to the region center as the representative.

In this paper, we evaluate the performance of the improved MDS algorithm with respect to the time and quality-of-retrieval parameters for different pivot selection options.

5 Implementation Results

We have implemented the priority optimal relaxation measure and employed it on a set of SMIL documents collected from the Web [78].

5.1 Observations on the Distance Metric

In this section, we first present a selected set of retrieval results to highlight the key observations to describe the properties of the proposed distance measures.

Figure 20 lists the media documents used for the evaluation and provides a comparison of these documents. Figure 21 shows the mapping between two of these temporal documents. Note that although most of the objects in both of these documents are the same, the temporal structure of the documents are quite different

Table 1 shows the distance values calculated using one of the proposed distance measures: priority-optimal relaxation and no-penalty for unmapped objects. Please refer to Figure 20 for the interpretation of these distance values. Although we do not present them in this paper, we would like to note that other metrics also have similar behaviors.

Ideally this matrix should be symmetric but we see a few anomalies (see O4 vs. O1sw and O1sw vs. O4). These anomalies are not due to the distance function, but due to the imperfections in the object mapping algorithm. Consider the following scenario: Let us say document D1 has object o1, o2. Let document D2 have object o3, o4. Let us assume that D1 is the query document. If o1 is similar to both o3 and o4 by a similar measure, we have tie between o2 and o3 during the mapping phase the algorithm picks the first object in the mapping set. Say, in this case, the mapping algorithm chooses o1 - o3 and o2 - o4. On the other hand, let us assume that when D2 is the query document, the mapping algorithm chooses o2 then we have a mapping such that o3 - o2 and o4 - o2. In such a scenario we see that the matrix is not symmetric. Our mapping algorithm suffers from this and hence the resulting anomalies. However, this can be corrected when it is possible to introduce a total order between objects that can be used in the ties.

The naming conventions used for the files as mentioned in [78]. The difference between *.smil and *c.smil files is that, in *.smil files images are black-and-white whereas in *c.smil files images have color. Since the corresponding temporal graphs are identical, the number of edges that need to be relaxed is 0.
The *sw.smil files introduce a *switch* statement such that depending on the bit rate available for download either the black and white images or the color images are displayed. Therefore, distances between O1.smil and O1sw.smil and O1c.smil and O1sw.smil should be 0. In addition, the complexity of the files and the number of objects increase as the sequence number increases. Thus O1*.smil series have fewer objects compared to O3*.smil compared to O5*.smil and so forth.

O1.smil vs. O2.smil The only difference between O1*.smil and O2*.smil series of documents is that, in the latter the *regions* tag is used for finer spatial control. Since the temporal structure has not changed, in temporal sense the files are identical. Hence the distance is 0.

O2.smil vs. O3.smil O3.smil has the region structure introduced in the O2 series, but in addition it also has additional image objects with temporal characteristics. Also, the duration for which some objects in the presentation are displayed has been increased. Consequently, the temporal structure has changed.

O3.smil vs. O4.smil The temporal structure has changed in this document. While in O3.smil a set of images were being displayed sequentially, now an audio object is presented in parallel to every image object. Thus, there is an increase in the number of objects in O4.smil when compared to O3.smil. The temporal information has changed slightly.

O4.smil vs. O5.smil The number of objects has increased as well as the temporal information has changed. Now text objects as well as audio objects are presented in parallel to image objects.

O5.smil vs. O6.smil Some images are now associated with anchors. An image that was presented as the last one of a sequence in O5.smil is now being presented as the first one in the sequence.

Figure 20: Documents used for observations

We see that, the distance values returned by the algorithm and the corresponding rankings match closely the intuitive temporal differences and relationships between these documents.

For the documents that we used, we have seen that distance calculation task took anywhere between couple of milliseconds to couple of hundred milliseconds. Note that this time depends on the number of objects in the merged documents, number of temporal constraints, and the amount of conflicts between two documents. Figure 22 shows the time complexity of the algorithm used for finding constraints to relax. More detailed information about this algorithm can be found in [2]. However, we note that most of the distance measure calculation should be done off-line, as it is the case in other media databases, and appropriate indexing and clustering algorithm should be used to improve the performance.

5.2 Indexing Experiment Setup

In order to observe the effects of varying clustering and indexing parameters, we run a set of experiments. In this section, we describe these experiments and evaluate the results. We collected 75 SMIL documents[1]. In order

[1]SMIL is an XML based authoring language for multimedia presentations on the Web.

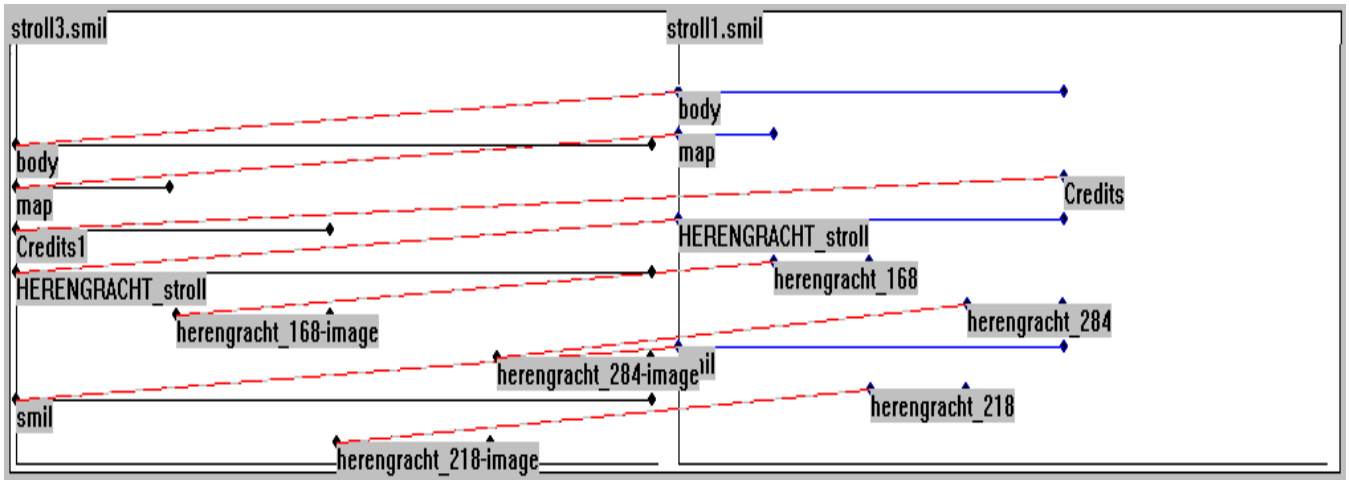


Figure 21: Example mapping between two documents: stroll3.smil (or O3.smil) and stroll1.smil (or O1.smil). Horizontal lines show the objects, whereas diagonal lines show the corresponding mappings.

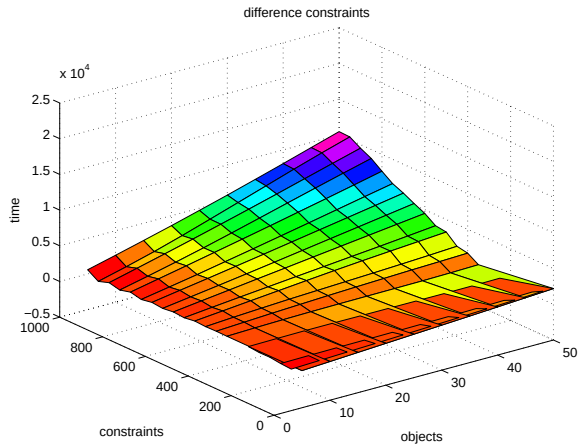


Figure 22: Time (in ms) required for distance calculation

to index these documents based on their temporal similarities, we applied the distance function we proposed in Section 5.1. Thus, in the preprocessing step, we created a 75×75 matrix, which describe the distances between each pair of documents. Note that this matrix is mainly used to evaluate the recall of the index alternative retrieval options.

We obtained a publicly available version of MDS from [79]. We choose 54 documents from the initial 75 documents and populated MDS structure with these documents. Thus, our observations are based on the case, where $N = 54$. The rest of the documents are used to observe the retrieval results for documents who are not in the database. With a stress value of 8 MDS indexed the selected 54 documents using a 6-dimensional vector space. To compare the retrieval quality of alternative retrieval options, we used a *recall* measure: We performed a top-10 retrieval and we compared the 10 results found using MDS with the 10 results that should have been returned if exhaustive search (full-database scan) was utilized. The motivation for using recall as the evaluation measure is that

	O1	O1c	O1sw	O2	O2c	O2sw	O3	O3c	O3sw	O4	O4c	O4sw	O5	O5c	O5sw	O6	O6c	O6sw
O1	0	0	0	0	0	0	13	13	13	13	13	13	13	13	13	14	14	14
O1c	0	0	0	0	0	0	13	13	13	13	13	13	13	13	13	14	14	14
O1sw	0	0	0	0	0	0	15	15	25	26	26	24	26	26	24	27	27	27
O2	0	0	0	0	0	0	13	13	13	13	13	13	13	13	13	14	14	14
O2c	0	0	0	0	0	0	13	13	13	13	13	13	13	13	13	14	14	14
O2sw	0	0	0	0	0	0	15	15	25	27	27	24	27	27	24	28	28	28
O3	13	13	15	13	13	15	0	0	0	7	7	5	8	8	8	13	13	14
O3c	13	13	15	13	13	15	0	0	0	7	7	5	8	8	8	13	13	14
O3sw	13	13	24	13	13	24	0	0	0	16	16	5	19	19	13	24	24	24
O4	13	13	27	13	13	27	7	7	14	0	0	14	17	17	17	25	25	26
O4c	13	13	27	13	13	27	7	7	14	0	0	14	17	17	17	25	25	26
O4sw	13	13	24	13	13	24	5	5	5	14	14	14	0	31	31	36	36	35
O5	13	13	26	13	13	27	8	8	19	17	17	31	0	0	0	34	34	34
O5c	13	13	26	13	13	27	8	8	19	17	17	31	0	0	0	34	34	34
O5sw	13	13	24	13	13	24	8	8	13	17	17	25	0	0	0	47	47	68
O6	14	14	27	14	14	28	13	13	24	25	25	36	34	34	47	0	0	4
O6c	14	14	27	14	14	28	13	13	24	25	25	36	34	34	47	0	0	4
O6sw	14	14	27	14	14	28	14	14	24	26	26	35	34	34	68	4	4	0

Table 1: Distance table for a set of SMIL objects (Appendix describes what these objects are and why they are related)

we want to know if the efficient MDS algorithm has the same quality of retrieval as the naive MDS. Note that another measure could be the absolute distance between the query point mapped using the efficient MDS and the naive MDS algorithms.

While this measure provides us with an error value such that if both the algorithms provide the same mapping then the error value would be 0, it does not provide us with any significant information in terms of the change in the quality of retrieval due to the diference in the mapping. In fact, we found during the course of our experiments, that if a point from the already mapped dataset is reintroduced as a query, then naive MDS results in a new mapping different from the originally mapped position. In effect, we find that an error is introduced by naive MDS even for points that are already mapped. In this context, we find that the absolute error value can not provide any useful information as to the quality of retrieval, although there would be a certain degree of correlation between the error values and recall values.

For the purposes of the evaluation, we define recall as:

Definition 5.1 (Recall) $Recall = D_r^m \div m$

- D_r^m *is the number of top* m *documents that are also retrieved using the modified MDS algorithm.*

5.3 Query Documents

For the purpose of evaluating our results, we chose six query documents. We selected these documents to observe effects of different data distributions: Out of the six query documents, we chose four which were already in the database: documents 2, 19, 25, 45. Note that document 25 is an outlier, whereas others have similar documents in the database. We also chose two documents that are outside of the database but in the original set of 75 documents.

Query Document	Recall rate(k=5)
19	1.0
2	1.0
25	0.9
45	1.0
55	1.0
67	0.9

Table 2: Recall with 5 handpicked pivot points for various query documents; recall is calculated with respect to top-10 document retrieval queries

Query Doument Number	$N^{'} = 54$	k=10	k=5
19	0.9	0.8	0.46
2	1.0	0.8	0.46
25	1.0	0.51	0.48
45	1.0	0.83	0.8
55	1.0	0.86	0.66
67	0.4	0.39	0.2

Table 3: Recall values for different query documents with m=10 using the random pivot selection strategy.

These query documents are 55 and 67. Thus the query set we used for evaluation is $Q = 2, 19, 25, 45, 55, 67$. Note that each experiment is repeated multiple times and the average result is presented.

5.4 Evaluation

The first step in our experiments is to confirm our intuition: it is possible to use less than N points to map a query point into the MDS space. For this purpose, we first handpicked a small set of the documents that would map a given query into the vector space in a way that will provide good recall values. Example recall value calculations are shown in Table 2. This table shows query documents mapped into the MDS space using 5 pivot documents instead of all 54 documents in the database. Note that the recall rate, as defined earlier, is quite high in most cases. This observation is valid for other query points as well.

Therefore, we make the following observations: It is possible to select a few documents to be used during query-to-space mapping in an intelligent manner so as to have a very high recall for top-10 document retieval. This shows that it is indeed possible to have good recall, without requiring all of the N objects in the database to calculate the spatial representation of a query object, q.

However, we note that while the above observation validates our approach of using only $k \leq N$ objects from the original set of N objects, it does not say how to choose the k objects to be used during the query mapping.

After observing that it is possible to use a few carefully selected objects to map queries with high recall rates, we wanted to observe whether such a careful selection is essential or not. For this purpose, we applied a *random selection strategy*, where we randomly choose a few points from the database and use these points to map queries into the space. Table 3 summarizes the recall values for 6 query documents for $m = 10$. We observe that the quality of the *random strategy* is not as good as the manual selection strategy; which means that there is a room for

Query Doument Number	$N' = 54$	k=16	k=10	k=5
19	0.9	0.93	0.9	0.48
2	1.0	0.93	0.98	0.46
25	1.0	0.7	0.54	0.44
45	1.0	1.0	0.96	0.82
55	1.0	1.0	0.95	0.94
67	0.4	0.1	0.49	0.33

Table 4: Recall values for different query documents with m=10 using the cluster based pivot selection strategy.

Query Doument Number	m=5	m=10	m=20
19	0.68	0.9	0.95
2	0.88	0.98	0.98
25	0.6	0.54	0.75
45	0.92	0.96	0.98

Table 5: Recall values using cluster-based strategy for different query documents with k=10 for different m.

improvement (i.e., more intelligent selection of points) in query-to-space mapping using less then N comparisons.

Cluster-based pivot selection strategy Next, we experimented to observe how much improvement a cluster based selection of points can provide. We applied the R-tree based clustering approach for the same set of query points and the results, are summarized in Table 4. In addition, the results were evaluated for varying top-m retrieval tasks ($m = 5, 10, 20$) for the clustered approach. The results are summarized in Table 5.

Spatial distribution strategy During our experiments, we observed that the recall rate depends both on the pivot points and the query document: for a given query, we can select different k clusters with different resulting recall rates; and a given selection of k clusters, gives different recall rates for different query points.

In order to better-understand the effect of spatial distribution of the pivot points, we ran a second set of experiments, where we tried to select the data points as representative of the MDS space as possible. The intution is to see, if an even coverage of the space would improve on the cluster representative-based coverage of the space. In order to observe the effect of the position of the pivots versus the size of the set of points they represent, we also experimented with weighted[2] and non-weighted use of pivots.

Table 6 shows the recall values for different strategies. Note that the spatial subdivision approach captures query documents that are rare (such as query document 25 with uniform spatial division) in the database, whereas the cluster-based approach captures query documents that are very similar to a large number of documents in the database. For the documents that are not in the database, the cluster-based approach performed a better query-to-space mapping.

For the uniform division, on the other hand, we see improvements in weighted measure over non-weighted measure. The non-weighted measure works better in capturing the outliers (such as document 25) as the spatial position of the pivots are more important then the number of points they represent in this case.

[2]We account for the number of documents represented by each pivot.

Query Document	$N' = 54$	cluster	random	uniform (no weight)	uniform
19	0.9	0.9	0.8	0.62	0.68
2	1.0	0.98	0.8	0.68	0.74
25	1.0	0.54	0.51	0.72	0.70
45	1.0	0.96	0.83	0.82	0.82
55	1.0	0.95	0.86	0.77	0.74
67	0.4	0.49	0.39	0.22	0.22

Table 6: Recall values for different query documents with k=10, m=10 using the different approaches.

5.5 Summary

In the light of the above observations, we can make the following conclusions

- The performance of the modified MDS can be as good as the standard MDS algorithm.
- Recall rate varies with the query point.
- In most cases, clustering provides better recall than the random and spatial subdivision strategies.
- As the number of documents to be retrieved is increases, the recall rate increases as well.
- Modified MDS performs better when there are documents similar to the query point in the database. This can be observed by looking at the recall values for query point 67, which is not in the database. The same also holds for query point 25 which is an outlier in the database. However we note that standard MDS also have a similar tendency as can be seen looking at query point 67.

6 Conclusions

In this paper, we described a similarity-based retrieval framework that addresses the challenges associated with the temporal nature of multimedia documents and temporal schedules. We described a system where users can query, not only the textual content of the media documents, but also the media objects contained within them as well as their temporal structures. For this purpose, we describe intuitive distance measures, which consider document authors' intentions, applicable for different retrieval tasks and we develop algorithms that efficiently compute these measures. We also developed algorithms that efficiently index temporal structures based on these measures. More specifically, we proposed a variant of the Multi Dimensional Scaling (MDS) algorithm and we showed that the proposed variant is both efficient and provides high quality retrieval results.

References

[1] K.S. Candan, B. Prabhakaran, and V.S. Subrahmanian. *A Framework for Supporting Multimedia Document Authoring and Presentation*, The Fourth ACM International Multimedia Conference '96, pp. 329-339, Boston, November 1996.

[2] K.S. Candan, B. Prabhakaran, and V.S. Subrahmanian. *Collaborative Multimedia Documents: Authoring and Presentation*, Special issue of the International Journal of Intelligent Systems on Multimedia Computing Systems, Volume 13(12), 1998.

[3] W.-S. Li, K.S. Candan, K. Hirata, and Y. Hara. *Hierarchical Image Modeling for Object-based Media Retrieval*, Data and Knowledge Engineering, Elsvier Science, Elsvier Science, Vol:27, Nr:2, pp. 139-176, July 1998.

[4] W.-S. Li, K.S. Candan, K. Hirata, and Y. Hara. *Supporting Efficient Multimedia Database Exploration*, VLDB Journal 9(4): 312-326, 2001.

[5] S. Adalı, K.S. Candan,Su-Shing Chen, K. Erol, and V.S. Subrahmanian. *Advanced Video Information System: Data Structures and Query Processing.* ACM-Springer Multimedia Systems Journal, Vol. 4, pp 172-186, August 96.

[6] F. Yaman, S. Adalı, D. Nau, M.L. Sapino, and V.S. Subrahmanian. *Plan Databases: Model and Algebra* submitted for publication.

[7] ISO/IEC 10744.

[8] World Wide Web Consortium (W3C). *Synchronized Multimedia Integration Language Recommendation*, 15 June, 1998.

[9] `http://www.w3.org/TR/REC-smil/`

[10] *The AthenaMuse 2 Functional Specification*, CECI(MIT) Report, May 1992.

[11] `http://www.macromedia.com/`

[12] ISO 13522-5.

[13] ISO 8879:1986.

[14] R. Hamakawa and J. Reikomoto. *Object Composition and Playback Models for Handling Multimedia Data* ACM Multimedia Conference '93, Pages 273-281, 1993.

[15] J.F. Allen. *Maintaining Knowledge about Temporal Intervals*, Communications of the ACM, vol. 26, no. 11, pp. 832-843, November 1983.

[16] J.F. Allen. *Towards a General Theory of Time and Action*, Artificial Intelligence, 23, pp. 123-154, 1984.

[17] T.D.C. Little and A. Ghafoor. *Synchronization and Storage Models for Multimedia Objects*, IEEE J. on Selected Areas of Communications, vol. 8, no. 3, April 1990, pp. 413-427. April 1990.

[18] T.D.C. Little and A. Ghafoor. *Interval-Based Conceptual Models for Time-Dependent Multimedia Data* IEEE Transactions on Knowledge and Data Engineering, vol. 5, no. 4, pp. 551-563, Aug 1993.

[19] B. Prabhakaran and S.V. Raghavan. *Synchronization Models for Multimedia Presentation With User Participation*, ACM/Springer-Verlag Journal of Multimedia Systems, vol.2, no. 2, August 1994, pp. 53-62. Also in the Proceedings of the First ACM Conference on Multimedia Systems, Anaheim, California, pp.157-166, August 1993.

[20] L. Li, A. Karmouch, and N.D. Georganas. *Multimedia Teleorchestra With Independent Sources: Part 1 and Part 2*, ACM/Springer-Verlag Journal of Multimedia Systems, vol. 1, no. 4, February 1994, pp.143-165. February 1994.

[21] World Wide Web Consortium (W3C). *Extensible Markup Language 1.0 Recommendation*, 10 Feb, 1998.

[22] M. Vilain and H. Kautz. *Constraint Propagation Algorithms for Temporal Reasoning*, Proceedings of AAAI-

86, Artificial Intelligence, pp. 377-382, August 1986.

[23] Claudio S. Pinhanez, Kenji Mase, and Aaron F. Bobick. *Interval Scripts: a Design Paradigm for Story-Based Interactive Systems.* CHI 1997, pp. 287-294, 1997.

[24] Claudio S. Pinhanez and Aaron F. Bobick. *Fast Constraint Propagation on Specialized Allen Networks and its Application to Action Recognition and Control*. Submitted to AAAI'98. Perceptual Computing Tech. Rep. No. 456. January 1998.

[25] K.S. Candan, B. Prabhakaran, and V.S. Subrahmanian. *Retrieval Schedules Based on Resource Availability and Flexible Presentation Specifications*, ACM-Springer Multimedia Systems Journal, Vol. 6, No. 4, pp. 232-250, 1998.

[26] M.C. Buchanan and P.T. Zellweger. *Automatic Temporal Layout Mechanisms* ACM Multimedia 93, pp. 341-350, 1993.

[27] M.C. Buchanan and P.T. Zellweger. *Scheduling Multimedia Documents Using Temporal Constraints*, Network and Operating System Support for Digital Audio and Video, P. Venkat Rangan (ed.), Springer-Verlag, Berlin Heidelberg New York, pp. 237-249, 1993.

[28] M.C. Buchanan and P.T. Zellweger. *Automatically Generating Consistent Schedules for Multimedia Documents*, ACM/Springer-Verlag Journal of Multimedia Systems, vol. 1, no. 2, 1993.

[29] M.Y. Kim and J. Song. *Hyperstories: Combining Time, Space and Asynchrony in Multimedia Documents*, RC19277(83726) (revised 1995) IBM Computer Science/Mathematics Research Report, 1993.

[30] M.Y. Kim and J. Song. *Multimedia Documents with Elastic Time*, ACM Multimedia Conference '95, 1995.

[31] J. Song, Y.N. Doganata, M.Y. Kim, and A.N. Tantawi. *Modeling Timed User-Interactions in Multimedia Documents*, IBM Computer Science Research Report, 1995. Also appeared in IEEE International Conference on Multimedia Computing and Systems '96, pp. 407-416, Hiroshima, June 1996.

[32] P. van Beek. *Approximation Algorithms for Temporal Reasoning*, Proceedings of the 11th International Joint Conference on Artificial Intelligence. Detroit, MI, USA, Morgan Kaufmann, pp. 1291-1296, August 1989.

[33] B. Aspavall and Y. Shiloach. *A Polynomial Time Algorithm for Solving Systems of Linear Inequalities with Two Variables per Inequality*, SIAM Journal of Computing, vol. 9, no. 4, pp. 827-845, November 1980.

[34] R. Dechter, I. Meiri, and J. Pearl. *Temporal Constraint Networks*, Artificial Intelligence, vol. 49, pp. 61-95, 1991.

[35] G. Özsoyoğlu, V. Hakkoymaz, and J.D. Kraft. *Automating the Assembly of Presentations from Multimedia Databases*, Twelfth International Conference on Data Engineering, pp. 593-601, New Orleans, February 1996.

[36] V. Hakkoymaz and G. Özsoyoğlu. *A Constraint-Driven Approach to Automate the Organization and Playout of Presentations in Multimedia Databases*, Journal of Multimedia Tools and Applications, Kluwer Pub., Vol. 4, pp. 171-197, 1997.

[37] V. Hakkoymaz, J. Kraft, and G. Özsoyoğlu. *Constraint-Based Automation of Multimedia Presentation Assembly*, ACM Multimedia Systems Journal, to appear, 1998.

[38] M. Vazirgiannis and S. Boll. *Events in Interactive Multimedia Applications: Modelling and Implementation Design*, IEEE International Conference on Multimedia Computing and Systems, June 1997.

[39] A. Del Bimbo, E. Vicario, and D. Zingoni. *Symbolic Description and Visual Querying of Image Sequences*

Using Spatio-Temporal Logic, TKDE, 7(4), pp. 609-622, 1995.

[40] C. Baral, G. Gonzalez, and T.C. Som. *Design and Implementation of Display Specifications for Multimedia Answers*, ICDE'98, pp. 558-565, Orlando, Florida, 1998.

[41] S. Wirag, *Scheduling of Adaptive Multimedia Documents*, ICMCS'99, pp. 307-311, June 1999.

[42] Y. Song, M. Mielke, and A. Zhang. *NetMedia: Synchronized Streaming of Multimedia Presentations in Distributed Environments*, ICMCS'99, pp. 585-590, June 1999.

[43] I. Mirbel, B. Pernici, and M. Vazirgiannis. *Temporal Integrity Constraints in Interactive Multimedia Documents*, ICMCS'99, pp. 867-871, June 1999.

[44] R.M. de Lima, P.A. da S. Gonçalves, F.P. Junqueria, and O.C.M.B. Duarte. *SAMM: An Integrated Environment to Support Multimedia Synchronization of Pre-orchestrated Data*, ICMCS'99, pp. 929-933, June 1999.

[45] S. Adalı, M.-L. Sapino, V.S. Subrahmanian. *A Multimedia Presentation Algebra*, SIGMOD'99, pp. 121-132, June 1999.

[46] M.L. Escobar-Molano, D.A. Barrett, Z. Genova, and L. Zhang. *Retrieval Scheduling for Multimedia Presentations*, Workshop on Multimedia Information Systems, November 7-9, 2001 - Capri, Italy.

[47] K.S. Candan. *A Framework for Distributed Multimedia Collaborations*, Ph.D. Thesis, University of Maryland at College Park.

[48] K.S. Candan, V.S. Subrahmanian, and P. Venkat Rangan. *Collaborative Multimedia Systems: Synthesis of Media Objects.* IEEE Trans. on Knowledge and Data Engineering, Vol. 10, Nr. 3, pp. 433-457, May-June 1998.

[49] K.S. Candan, V.S. Subrahmanian, and P.V. Rangan. *Towards a Theory of Collaborative Multimedia*, IEEE International Conference on Multimedia Computing and Systems, pp. 279-282, Hiroshima, Japan, June 96.

[50] K.S. Candan and R. Cutler. *Multimedia Authoring Systems.* Appeared as a chapter in the book *Multimedia Database Systems* published by Springer-Verlag,(eds. V.S. Subrahmanian and S. Jajoida).

[51] K.S. Candan, E. Hwang, and V.S. Subrahmanian. *An Event-Based Model for Continuous Media Data on Heterogeneous Disk Servers.* Accepted for publication in ACM-Springer Multimedia Systems Journal.

[52] K.S. Candan, E. Hwang, B. Prabhakaran, and V.S. Subrahmanian. *Resource Lock Commit Protocol (RLCP) for Multimedia Object Retrieval*, Technical Report, UMCP-CSD CS-TR-3746, UMIACS-TR-97-08, February 1997.

[53] K.S. Candan, B. Prabhakaran, and V.S. Subrahmanian. *Collaborative Multimedia Documents: Authoring and Presentation*, Accepted for publication in the Special Issue of the International Journal of Intelligent Systems on Multimedia Computing Systems.

[54] K.S. Candan, E. Lemar, and V.S. Subrahmanian. *Management and Rendering of Multimedia Views*, Fourth International Workshop on Multimedia Information Systems, 1998.

[55] K.S. Candan, E. Lemar, V.S. Subrahmanian. *View Management in Multimedia Databases.* VLDB Journal 9(2): 131-153, 2000.

[56] D. Suciu, *Semistructured Data and XML*, Proceedings of 5th International Conference of Foundations of Data Organization (FODO'98), Kobe, Japan, November 1998.

[57] K.-C. Tai. *The Tree-to-Tree Correction Problem*, Journal of ACM, pp. 422-433, vol 26, March 1979.

[58] K. Zhang and D. Shasha. *Simple Fast Algorithms for the Editing Distance between Trees and Related Prob-*

lems, SIAM Journal of Computing, 18(6), pp. 1245-1262, 1989.

[59] D. Shasha and K. Zhang. *Fast Algorithms for the Unit Cost Distance between Trees*, Journal of Algorithms, 11, pp. 581-621, 1990.

[60] K. Zhang, J.T.L. Wang, and D. Shasha. *On the Editing Distance between Undirected Acyclic Graphs*, International Journal of Computer Science, 7(1), pp. 43-57, 1996.

[61] S. Chawathe. *Comparing Hierarchical Data in External Memory*, to appear in Proceedings of the Twenty-fifth International Conference on Very Large Data Bases, Edinburgh, Scotland, U.K., 1999.

[62] S. Chawathe and H. Garcia-Molina. *Meaningful Change Detection in Structured Data*, Proceedings of the ACM SIGMOD International Conference on Management of Data , pp. 26-37, Tucson, Arizona, May 1997.

[63] S. Selkow. *The Tree-to-tree Editing Problem*, Information Processing Letters, 6(6), pp. 184-186, 1977.

[64] F. Luccio and L. Pagli. *Approximate Matching for Two Families of Trees*, Information and Computation, 123(1), pp. 111-120,1995.

[65] M. Farach and M. Thorup. *Sparse Dynamic Programming for Evolutionary-tree Comparison*, SIAM Journal on Computing, 26(1):210-230, January 1997.

[66] E. Myers. *An $O(ND)$ Difference Algorithms and its Variations.* Algorithmica, 1(2), pp. 251-266,1986.

[67] K.S. Candan, W.-S. Li, M.L. Priya. *Similarity-based Ranking and Query Processing in Multimedia Databases.* DKE 35(3): 259-298, 2000.

[68] K.S. Candan, W.-S. Li. *On Similarity Measures for Multimedia Database Applications.* Knowledge and Information Systems 3(1): 30-51, 2001.

[69] Prakash Yamuna and K.S. Candan. *Similarity based retrieval of Temporal Documents*,International Workshop on Multimedia Information Retrieval, Nov. 2000.

[70] S. Nestorov, J. Ullman, J. Wiener, and S. Chawathe. *Representative Objects: Concise Representations of Semistructured, Hierarchical Data*, in Proceedings of the International Conference on Data Engineering, pp. 79-90, U.K., 1997.

[71] C. Faloutosos and Kin-Ip (David) Lin. *Fast Map: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets*, ACM SIGMOD, May 1995, San Jose, CA, pp.163-174.

[72] J.B. Kruskal. *Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis*, Psychometrika, 29(1):1-27, March 1964.

[73] J.B. Kruskal. *Nonmetric MultiDimensional Scaling: A Numerical Method*, Psychometrika, 29(2):115-129, June 1964.

[74] R. Fagin and L.Stockmeyer.*Relaxing the Triangle Inequality in Pattern Matching*, International Journal of Computer Vision, 28, 3, 1998, pp.219-231.

[75] W.S. Torgerson. *Multidimensional scaling: I. theory and method*, Psycometrika, 17:401-419, 1952.

[76] J.B. Kruskal and Myron Wish. *Multidimensional scaling*, SAGE publications, Beverly Hills, 1978.

[77] P. Yamuna and K. S. Candan. *Efficient Similarity-based Retrieval of Temporal Structures*, Symposium on Applications and the Internet, Jan. 2001.

[78] *www.oratrix.com/GRiNS/gse/index.html*, available as examples of GRiNS Version 1.0 for SMIL 1.0 player

[79] *http://netlib.bell-labs.com/netlib/master/readme.html*