A Rule-Based Approach to Represent Spatio-Temporal Relations in Video Data*

Mehmet E. Dönderler, Özgür Ulusoy, and Uğur Güdükbay

Department of Computer Engineering,
Bilkent University, Bilkent, 06533 Ankara, Turkey
{mdonder, oulusoy, gudukbay}@cs.bilkent.edu.tr

Abstract. In this paper, we propose a new approach for high level segmentation of a video clip into shots using spatio-temporal relationships between objects in video frames. The technique we use is simple, yet novel and powerful in terms of effectiveness and user query satisfaction. Video clips are segmented into shots whenever the current set of relations between objects changes and the video frames where these changes have occurred are chosen as key frames. The topological and directional relations used for shots are those of the key frames that have been selected to represent shots and this information is kept, along with key frame intervals, in a knowledge-base as Prolog facts. We also have a comprehensive set of inference rules in order to reduce the number of facts stored in our knowledge-base because a considerable number of facts, which otherwise would have to be stored explicitly, can be derived by these rules with some extra effort.

Keywords: video modeling, rule-based systems, video databases, spatio-temporal relations.

1 Introduction

There is an increasing demand toward multimedia technology in recent years. Especially, first image and later video databases have attracted great deal of attention. Some examples of video search systems developed thus far are OVID [9], Virage [5], VideoQ [2] and VIQS [6].

One common property of the image and video databases is the existence of spatial relationships between salient objects. Besides, video data also has a time dimension, and consequently, objects change their locations and their relative positions with respect to each other in time. Because of this, we talk about spatio-temporal relationships rather than spatial or temporal relationships alone for video data. A spatio-temporal relationship between two objects can be defined on an interval of video frames during which the relation holds.

* This work is supported by the Scientific and Research Council of Turkey (TÜBİTAK) under Project Code 199E025.

T. Yakhno (Ed.): ADVIS 2000, LNCS 1909, pp. 409–418, 2000.

This paper is concerned with the representation of topological and directional relations within video data. We propose a new approach for high level video segmentation based on the spatio-temporal relationships between objects in video. Video clips are segmented into shots whenever the current set of topological and directional relations between video objects changes, thereby helping us to determine parts of the video where the spatial relationships do not change at all. Extraction of the spatio-temporal relations and detection of the key frames for shots are part of our work as well.

We believe that this high level video segmentation technique results in an intuitive and simple representation of video data with respect to spatio-temporal relationships between objects and provides more effective and precise answers to such user queries that involve objects' relative spatial positions in time dimension. Selecting the key frames of a video clip by the methods of scene detection, as has been done in all systems we have looked into so far, is not very well suited for spatio-temporal queries that require searching the knowledge-base for the parts of a video where a set of spatial relationships holds and does not change at all. For example, if a user wishes to query the system to retrieve the parts of a video clip where two persons shake their hands, the video fragments returned by our system will have two persons shaking hands at each frame when displayed. However, other systems employing traditional scene (shot) detection techniques would return a superset where there would most probably be other frames as well in which there is no handshaking at all. The reason is that current methods of scene detection mainly focus on the camera shots rather than the change of spatial relationships in video.

We use a rule-based approach in modeling spatio-temporal relationships. The information on these relationships is kept in our knowledge-base as Prolog facts and only the basic relations are stored whilst the rest may be derived in the process of a query using the inference rules we provide by using Prolog. In our current implementation, we keep a single key frame interval for each fact, which reduces the number of facts stored in the knowledge-base considerably.

The organization of this paper is as follows: In Sect. 2, we describe and give the definitions for spatio-temporal relations. Our rule-based approach to represent topological and directional relations between video salient objects, along with our inference rule definitions, is introduced in Sect. 3. Section 4 gives some example queries based on an imaginary soccer game whereby we demonstrate our rule-based approach. We briefly mention about our performance experiments in Sect. 5. Finally, we present our conclusions in Sect. 6.

2 Spatio-Temporal Relationships

The ability to manage spatio-temporal relationships is one of the most important features of the multimedia database systems. In multimedia databases, spatial relationships are used to support content-based retrieval of multimedia data, which is one of the most important differences in terms of querying between multimedia and traditional databases. Spatial relations can be grouped into mainly three

categories: topological relations, which describe neighborhood and incidence, directional relations, which describe order in space, and distance relations that describe range between objects. There are eight distinct topological relations: *disjoint*, *touch*, *inside*, *contains*, *overlap*, *covers*, *covered-by* and *equal*. The fundamental directional relations are *north*, *south*, *east*, *west*, *north-east*, *north-west*, *south-east* and *south-west*, and the distance relations consist of *far* and *near*. We also include the relations *left*, *right*, *below* and *above* in the group of directional relations; nonetheless, the first two are equivalent to the relations *west* and *east*, and the other two can be defined in terms of the directional relations as follows:

Above The relation *above*(A,B) is the disjunction of the directional relations *north*(A,B), *north-west*(A,B) and *north-east*(A,B).

Below The relation *below*(A,B) is the disjunction of the directional relations *south*(A,B), *south-west*(A,B) and *south-east*(A,B).

Currently, we only deal with the topological and directional relations and leave out the distance relations to be incorporated into our system in future. We give our formal definitions for the fundamental directional relations in Sect. 2.1. The topological relations are introduced in Sect. 2.2. Further information about the topological and directional relations can be found in [4,7,8,10]. Finally, in Sect. 2.3, we explain our approach to incorporate the time component into our knowledge-base to facilitate spatio-temporal and temporal querying of video data.

2.1 Directional Relations

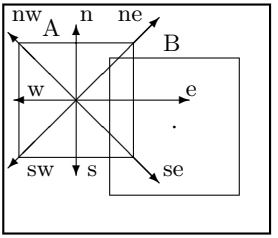
To determine which directional relation holds between two salient objects, we consider the center points of the objects' minimum bounding rectangles (MBRs). Obviously, if the center points of the objects' MBRs are the same, then there is no directional relation between the two objects. Otherwise, we choose the most intuitive directional relation with respect to the closeness of the line segment between the center points of the objects' MBRs to the eight directional line segments. To do this, we place the origin of the directional system at the center of the MBR of the object for which to define the relation as illustrated in Fig. 1(a).

Even if two objects overlap with each other, we can still define a directional relation between them. In other words, objects do not have to be disjoint to define a directional relation between as opposite to the work of Li et al [8]. Our approach to find the directional relations between two salient objects can be formally expressed as in Definitions 1 and 2.

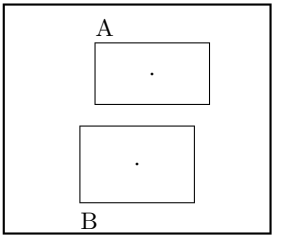
Definition 1. *The directional relation* β(A,B) *is defined to be in the opposite direction to the directional line segment which originates from the center of object* A *'s* MBR *and is the closest to the center of object* B *'s* MBR*.*

Definition 2. *The inverse of a directional relation* β(A, B), β^{-1}(B,A)*, is the directional relation defined in the opposite direction for the objects* A *and* B*.*

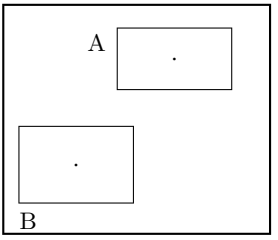
Examples of the fundamental directional relations are illustrated in Fig. 1.



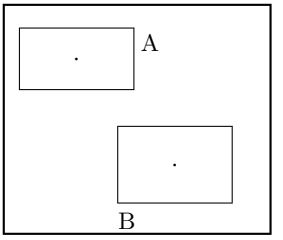
(a) west(A,B), east(B,A)



(b) north(A,B), south(B,A)



(c) north-east(A,B), south-west(B,A)



(d) north-west(A,B), south-east(B,A)

Fig. 1. Directional Relations

2.2 Topological Relations

The topological relations *inside* and *contains* are inverses of each other, and so are *cover* and *covered-by*. In addition, the relations *equal*, *touch*, *disjoint* and *overlap* hold in both directions. In other words, if β(A,B) holds where β is one of these four relations, then β(B,A) holds too.

The topological relations are distinct from each other; however, the relations *inside*, *cover* and *equal* imply the same topological relation *overlap* to hold between the two objects:

- *inside*(A,B) $\Longrightarrow$ *overlap*(A,B) $\wedge$ *overlap*(B,A)
- *cover*(A,B) $\Longrightarrow$ *overlap*(A,B) $\wedge$ *overlap*(B,A)
- *equal*(A,B) $\Longrightarrow$ *overlap*(A,B) $\wedge$ *overlap*(B,A)

We base our definitions for the topological relations on Allen's temporal interval algebra [1] and Fig. 2 gives some examples of the topological relations.

2.3 Temporal Relations

We use time intervals to model the time component of video data. All directional and topological relations for a video have a time component, a time interval specified by the starting and ending frame numbers, associated with them during which the relations hold. With this time component attached, relations are not

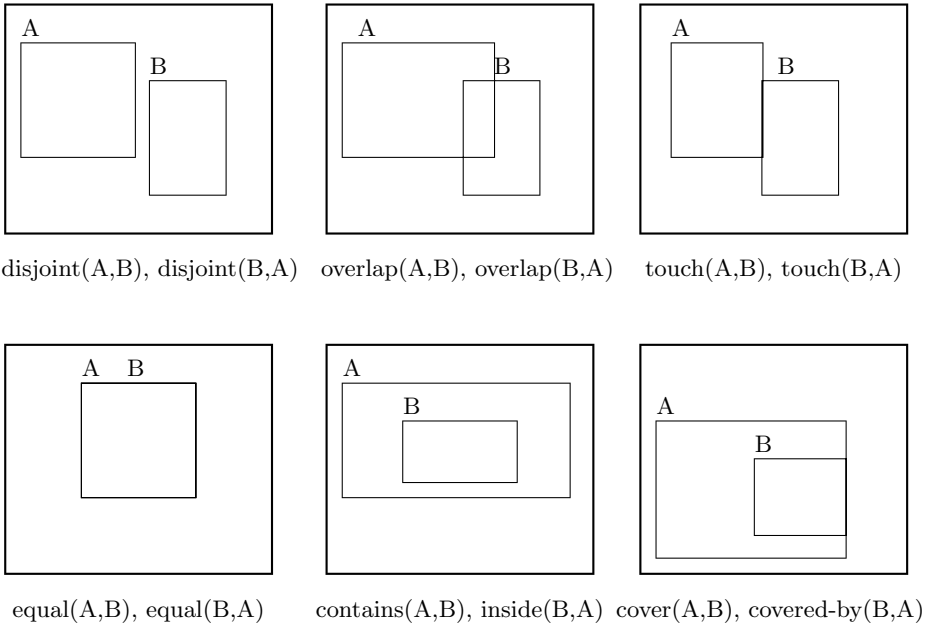


Fig. 2. Topological Relations

anymore simple spatial relations, but rather spatio-temporal relations that we use to base our model upon.

The topic of relations between temporal intervals has been addressed and discussed in [1]. There are seven temporal relations: *before*, *meets*, *during*, *overlaps*, *starts*, *finishes* and *equal*. Inverses of these temporal relations are also defined and the inverse of the temporal relation *equal* is itself.

3 A Rule-Based Approach for Spatio-Temporal Relations

Rules have been extensively used in knowledge representation and reasoning. The reason why we have employed a rule-based approach to model the spatio-temporal relations between salient objects is that it is very space efficient.

We use Prolog for rule processing and querying of spatio-temporal relations. By rules, we can derive some spatial knowledge which is not explicitly stored within the database; thus, we only store some basic facts in our knowledge-base and let Prolog generate the rest of the knowledge itself. Besides, our rule-based approach provides an easy-to-process and easy-to-understand knowledge-base structure for our video search system.

Our approach greatly reduces the number of relations to be stored in the knowledge-base which also depends on some other factors as well, such as the number of salient objects, the frequency of change in spatial relations and the relative spatial locations of the objects with respect to one another. Nevertheless,

we do not currently claim that the set of relations we store in our knowledge-base is a minimum set of facts that must be stored, but we have been working on improving our algorithm considering the dependencies among our rules.

In our system, we define three types of inference rules, *strict directional*, *strict topological* and *heterogeneous directional and topological*, with respect to the relations' types in the rule body. For example, *directional rules* have only directional relations in their body whilst *heterogeneous rules* incorporate rules from both types.

We describe our *strict directional rules* and *strict topological rules* in Sects. 3.1 and 3.2, respectively. Our *heterogeneous topological and directional rules* are given in Sect. 3.3. More elaborate discussion of our rule-based approach can be found in [3].

In defining the rules, we have adopted the following terminology: if the relation r_1 implies the relation r_2, we show it by $r_1 \Longrightarrow r_2$. Moreover, if $r_1 \Longrightarrow r_2$ and $r_2 \Longrightarrow r_1$, we denote it by $r_1 \Longleftrightarrow r_2$.

3.1 Strict Directional Rules

Rule 1 (Inverse Property) The directional relations *west*, *north*, *north-west*, *north-east*, *right* and *above* are inverses of *east*, *south*, *south-east*, *south-west*, *left* and *below*, respectively.

$$\textit{west}(\mathrm{A},\mathrm{B}) \Longleftrightarrow \textit{east}(\mathrm{B},\mathrm{A})$$
$$\textit{north}(\mathrm{A},\mathrm{B}) \Longleftrightarrow \textit{south}(\mathrm{B},\mathrm{A})$$
$$\textit{north-west}(\mathrm{A},\mathrm{B}) \Longleftrightarrow \textit{south-east}(\mathrm{B},\mathrm{A})$$
$$\textit{north-east}(\mathrm{A},\mathrm{B}) \Longleftrightarrow \textit{south-west}(\mathrm{B},\mathrm{A})$$
$$\textit{right}(\mathrm{A},\mathrm{B}) \Longleftrightarrow \textit{left}(\mathrm{B},\mathrm{A})$$
$$\textit{above}(\mathrm{A},\mathrm{B}) \Longleftrightarrow \textit{below}(\mathrm{B},\mathrm{A})$$

Rule 2 (Transitivity) If $\beta \in \mathrm{S}$, where S is the set of directional relations, then $\beta(\mathrm{A},\mathrm{B}) \wedge \beta(\mathrm{B},\mathrm{C}) \Longrightarrow \beta(\mathrm{A},\mathrm{C})$.

3.2 Strict Topological Rules

The *strict topological rules* can be formally described as follows:

Rule 1 (Inverse Property) The topological relations *inside* and *cover* are inverses of *contains* and *covered-by*, respectively.

$$\textit{inside}(\mathrm{A},\mathrm{B}) \Longleftrightarrow \textit{contains}(\mathrm{B},\mathrm{A})$$
$$\textit{cover}(\mathrm{A},\mathrm{B}) \Longleftrightarrow \textit{covered-by}(\mathrm{B},\mathrm{A})$$

Rule 2 (Reflexivity) The topological relations *equal* and *overlap* are reflexive.

$$\textit{equal}(\mathrm{A},\mathrm{A}),\ \textit{overlap}(\mathrm{A},\mathrm{A})$$

Rule 3 (Symmetry) The topological relations *equal*, *overlap*, *disjoint* and *touch* are symmetric.

$equal(A,B) \iff equal(B,A)$
$overlap(A,B) \iff overlap(B,A)$
$disjoint(A,B) \iff disjoint(B,A)$
$touch(A,B) \iff touch(B,A)$

Rule 4 (Transitivity) The topological relations *inside* and *equal* are transitive.

$inside(A,B) \wedge inside(B,C) \Longrightarrow inside(A,C)$
$equal(A,B) \wedge equal(B,C) \Longrightarrow equal(A,C)$

Rule 5 The topological relations *inside*, *equal* and *cover* imply the relation *overlap*.

$inside(A,B) \Longrightarrow overlap(A,B)$
$equal(A,B) \Longrightarrow overlap(A,B)$
$cover(A,B) \Longrightarrow overlap(A,B)$

Rule 6 The relationships between *equal* and {*cover*, *inside*, *disjoint*, *touch*, *overlap*} are as follows:

a) $equal(A,B) \wedge cover(A,C) \Longrightarrow cover(B,C)$
b) $equal(A,B) \wedge cover(C,A) \Longrightarrow cover(C,B)$
c) $equal(A,B) \wedge inside(A,C) \Longrightarrow inside(B,C)$
d) $equal(A,B) \wedge inside(C,A) \Longrightarrow inside(C,B)$
e) $equal(A,B) \wedge disjoint(A,C) \Longrightarrow disjoint(B,C)$
f) $equal(A,B) \wedge overlap(A,C) \Longrightarrow overlap(B,C)$
g) $equal(A,B) \wedge touch(A,C) \Longrightarrow touch(B,C)$

Rule 7 The relationships between *disjoint* and {*inside*, *touch*} are as follows:

a) $inside(A,B) \wedge disjoint(B,C) \Longrightarrow disjoint(A,C)$
b) $inside(A,B) \wedge touch(B,C) \Longrightarrow disjoint(A,C)$

Rule 8 The relationships between *overlap* and {*inside*, *cover*} are as follows (excluding those given by Rule 5):

a) $inside(B,A) \wedge overlap(B,C) \Longrightarrow overlap(A,C)$
b) $cover(A,B) \wedge overlap(B,C) \Longrightarrow overlap(A,C)$

Rule 9 The relationships between *inside* and *cover* are as follows:

a) $inside(A,B) \wedge cover(C,B) \Longrightarrow inside(A,C)$
b) $inside(A,C) \wedge cover(A,B) \Longrightarrow inside(B,C)$
c) $cover(A,B) \wedge cover(B,C) \wedge \text{not}(\text{inside}(C,A)) \Longrightarrow cover(A,C)$

3.3 Heterogeneous Topological and Directional Rules

Rule 1 If $\beta \in S$, where S is the set of directional relations, then
$equal(A,B) \wedge \beta(A,C) \Longrightarrow \beta(B,C)$.

Rule 2 If $\beta \in S$, where S is the set of directional relations, then
$disjoint(A,B) \wedge disjoint(B,C) \wedge \beta(A,B) \wedge \beta(B,C) \Longrightarrow disjoint(A,C)$.

4 Query Examples

This section provides three query examples based on an imaginary soccer game fragment between England's two soccer teams, *Liverpool* and *Manchester United*. More query examples, along with the results returned by the system, can be found in [3].

Query 1 "Give the number of shots to the goalkeeper of *Liverpool* by each player of *Manchester United* and for the team *Manchester United* as a whole".

In this query, we are interested in the shots to the goalkeeper of *Liverpool* by each player of *Manchester United* except for the goalkeeper. The total number of shots to the goalkeeper of *Liverpool* by the team *Manchester United* will also be displayed.

We find the facts of *touch* to the ball for each player of *Manchester United* except for the goalkeeper. For each fact found, we also check if there is a fact of *touch* to the ball for the opponent team's goalkeeper, whose time interval comes after. Then, we check if there is no other *touch* to the ball between the intervals of the two facts and also if the ball is inside the field during the whole event. If all above is satisfied, this is considered a shot. Then, we count all such events to find the total number of shots to the goalkeeper by each *Manchester United* team member. The total number of shots to the goalkeeper of *Liverpool* by the team *Manchester United* is also computed.

Query 2 "Give the average ball control (play) time in frames for each player of *Manchester United*".

As we assume that when a player touches the ball, it is in his control, we calculate the ball control time for a player with respect to the time interval during which he is in touch with the ball. The average ball control time for a player is simply the sum of all time intervals where the player is in touch with the ball divided by the number of these time intervals. We could also give the time information in seconds provided that the frame rate of the soccer video is known.

To answer this query, we find for each player of *Manchester United*, except for the goalkeeper, the time intervals during which the player touches the ball and sum up the number of frames in the intervals. Divided by the number of facts found for each player, this gives us for each player of *Manchester United* the average ball control time in frames. Since in a soccer game, a player may touch the ball outside the field as well, we consider only the time intervals when the ball is inside the field.

Query 3 "Give the number of kicks outside the field for *David Beckham* of *Manchester United*".

We first find the time intervals when *David Beckham* of *Manchester United* is in touch with the ball while the ball is inside the field. Then, for each time

interval found, we look for a fact, whose time interval comes after, representing the ball being outside the field. If there is no *touch* to the ball between these two intervals, then this is a kick outside the field. We count all such occasions to find the total number of kicks outside the field by *David Beckham*.

5 Performance Experiments

We have tested our rule-based system for its performance using some randomly generated data. In conducting these tests, two criteria have been considered: space and time efficiency.

For the space efficiency part, we have looked into how well our system performs in reducing the number of redundant facts. The results show that our inference rules provide considerable improvements in space as the number of salient objects per frame increases. For an example of 1000-frame randomly generated video data where there are 50 objects at each frame, our savings is 53.15%. The ratio for an example with 25 objects at each frame is 40.42%. We have also observed that the space savings ratio is not dependent on the number of frames, but rather the number of salient objects per frame. In addition, we are also certain that our rule-based system will perform better in space efficiency tests if our fact-extraction algorithm is enhanced with a more successful fact-reduction feature.

For the time efficiency part, we have seen that our system is scalable in terms of the number of objects and the number of frames when either of these numbers is increased while the other is fixed.

Unfortunately, we are not able to present our performance test results in detail in this paper due to lack of space.

6 Conclusions

We presented a novel approach to segment a video clip using spatio-temporal relationships assuming that the topological and directional relations of each frame have already been extracted. We extract the topological and directional relations by manually specifying the objects' MBRs and detect the key frames for shots using this relationship information.

In our approach, whenever the current set of directional and topological relations between salient objects changes, we define a new key frame and use that frame as a representative frame for the interval between this key frame and the next one where the spatial relations are the same.

We use a knowledge-base to store the spatio-temporal relations for querying of video data. The knowledge-base contains a set of facts to describe some basic spatio-temporal relations between salient objects and a set of inference rules to infer the rest of the relations that is not explicitly stored. By using inference rules, we eliminate many redundant facts to be stored in the knowledge-base since they can be derived with some extra effort by rules.

We have also tested our system using some randomly generated data for the space and time efficiency measures. The results show that our rule-based system is scalable in terms of the number of objects and the number of frames, and that the inference rules provide considerable space savings in the knowledge-base.

Currently, we are developing the graphical user interface part of our WEB-based video search system. Users will be able to query the system using animated sketches. A query scene will be formed as a collection of objects with different attributes. Attributes will include motion, spatio-temporal ordering of objects and annotations. Motion will be specified as an arbitrary polygonal trajectory with relative speed information for each query object. Annotations will be used to query the system based on keywords. There will also be a category grouping of video clips in the database so that a user is able to browse the video collection before actually posing a query.

References

1. J.F. Allen. Maintaining knowledge about temporal intervals. *Communications of ACM*, 26(11):832–843, 1983.
2. S. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong. Videoq: An automated content-based video search system using visual cues. In *ACM Multimedia*, Seattle, Washington, USA, 1997.
3. M.E. Donderler, O. Ulusoy, and U. Gudukbay. Rule-based modeling of spatio-temporal relations in video databases. *Journal Paper in Preparation*.
4. M. Egenhofer and R. Franzosa. Point-set spatial relations. *Int'l Journal of Geographical Information Systems*, 5(2):161–174, 1991.
5. A. Hampapur, A. Gupta, B. Horowitz, C-F. Shu, C Fuller, J. Bach, M. Gorkani, and R. Jain. Virage video engine. In *SPIE*, volume vol. 3022, 1997.
6. E. Hwang and V.S. Subrahmanian. Querying video libraries, June 1995.
7. J.Z. Li and M.T. Özsu. Point-set topological relations processing in image databases. In *First International Forum on Multimedia and Image Processing*, pages 54.1–54.6, Anchorage, Alaska, USA, 1998.
8. J.Z. Li, M.T. Özsu, and D. Szafron. Modeling of video spatial relationships in an object database management system. In *Proceedings of the International Workshop on Multimedia DBMSs*, pages 124–133, Blue Mountain Lake, NY, USA, 1996.
9. E. Oomoto and K. Tanaka. OVID: Design and implementation of a video object database system. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):629–643, 1993.
10. D. Papadias, Y. Theodoridis, T. Sellis, and M. Egenhofer. Topological relations in the world of minimum bounding rectangles: A study with R-trees. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 92–103, San Jose, CA, USA, 1996.