# QUERYING VIDEO DATA BY SPATIO-TEMPORAL RELATIONSHIPS OF MOVING OBJECT TRACES

Chikashi Yajima† Yoshihiro Nakanishi† Katsumi Tanaka††

† *Division of Computer and Systems Engineering, Graduate School of Science and Technology, Kobe University. 1-1, Rokkodai, Nada, Kobe, Hyogo 657-8501, Japan*

{yajima, nakanisi } @db.cs.kobe-u.ac.jp

†† *Department of Social Informatics, Graduate School of Informatics, Kyoto University. Yoshida Honmachi, Sakyo, Kyoto 606-8501, Japan*

ktanaka@i.kyoto-u.ac.jp

**Abstract** We propose a query method that allows viewers to search video data based on the spatio-temporal relationships of moving objects using a simple and intuitive mode of input. In order to make a query, for example, of a certain segment or style of play in sports footage, it is necessary to express the complex movements of objects in an intuitive manner. In the proposed method, the user inputs this information as an object trajectory using a mouse or pen via a GUI. The usability of our proposed method is evaluated using a prototype query system with four main function; video query based on object speed, spatially absolute or relative query based on object movement, querying of movements of multiple moving objects specifying the spatio-temporal relationships among objects by expressing each object's trace on a *timeline* or *multi-canvas*, and providing a non-linear viewing experience by allowing the user to control playback by drawing object traces directly on the video screen while watching the video.

**Keywords:** video data, spatio-temporal, direct specification, moving object, trace

## 1. INTRODUCTION

Recently, much attention has been given to retrieving and delivering video data because of the rapid advances in digital video technologies. In order to retrieve video data, it is necessary to attach indices of meta data to the contents of the video data. For example, meta data can be automatically computed from video as feature vectors or manually

X. Zhou et al. (eds.), *Visual and Multimedia Information Management*

inserted by annotators as keyword indices. The user can use this meta data in order to retrieve video intervals. However, it should be noted that there are events that cannot be expressed by keywords. For example, as there is a limited vocabulary to represent movement, it is impossible to retrieve a scene that satisfies the following request: "I want to see images of a player moving *in this way.*" In this paper, we consider how video data for multiple moving objects can be queried.

In order to make that query, it is necessary to be able to express an object's complex movements or the spatio-temporal relationships between objects. To define such a query language that can describe various movements, the continuously changing positions of multiple moving objects have to be described. However, the query language seems to be so complex that users might not easily be able to make that query.

We propose a method for querying video intervals by drawing moving object traces with a mouse or a pen via a graphical user interface (GUI). This method is intuitive in that it does not require special background knowledge or training. The user draws traces and then uses the traces to search for video intervals that capture similar object movements. In order to verify the usability of our proposed method, we constructed a prototype. The prototype has the following four main functions; querying video images with consideration of an object's speed to some extent,

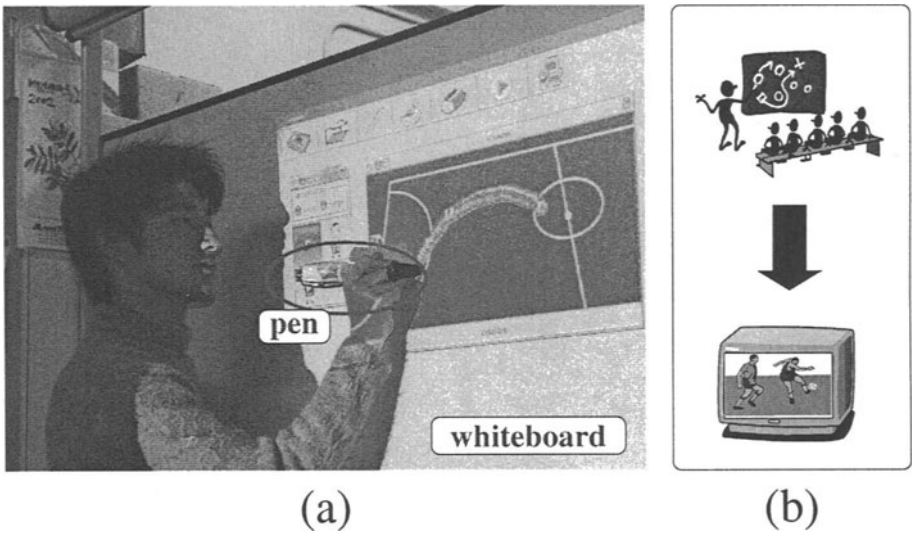


*Figure 1* (a) Demonstration of our prototype (b) An example of assumed application

spatially absolute or relative querying of an object's movement, querying of movements of multiple moving objects specifying spatio-temporal relationships among objects by expressing each object's trace on a *timeline* or *multi-canvas*, and controlling playback to provide a non-linear television experience by drawing moving object traces directly on a video image while watching a video.

Figure 1(a) shows a demonstration of the prototype we developed. The prototype allows us to make queries by drawing moving object traces with a pen on a whiteboard. For video database systems that contain a time-series of positional data of captured objects, our approach has several applications. One is a type of video on demand (VOD) application, in which the user can easily specify his favorite scenes by drawing traces. The other is a type of analysis application that is used for sport video data (See Figure 1(b)). Hereafter, we assume that video data contains time-series data indicating the positions of moving objects as meta data. Due to recent advances in positioning systems, this data can be generated by using small GPS-based sensors.

The paper is organized as follows. Section 2 provides a survey on related work. In Section 3 and 4 we discuss the method for querying video intervals based on the spatio-temporal relationships of moving object traces. In Section 5 we describe a query based on drawing of moving object traces directly onto a video screen while watching a video. Finally, we conclude this paper in Section 6.

## 2. RELATED WORK

There is growing interest in querying the large resources of digital video data. Allen laid the foundation for multiple research studies on time intervals (Allen, 1983). He showed that there are 13 distinct temporal relationships that can exist between two arbitrary time intervals. In our work, we propose a method to query video intervals by querying the movements of multiple moving objects. Here, it is necessary to specify not only the temporal relationships but the spatial relationships as well. We therefore propose a *timeline* that can specify both the 13 temporal relationships and the spatial relationships.

Generally, the most well-known method of querying video data is by using keywords. Previous work (Oomoto and Tanaka, 1993) (Hwang and Subrahmanian, 1996) emphasized the annotation model in order to efficiently retrieve video intervals indexed by a set of keywords. The Informedia project (Wactlar et al., 1996) uses the full-text information retrieval system for queries based on keywords for video databases. However, our prototype does not assume the existence of such keywords.

Meta data and queries in our work are a time-series of position and drawn moving object traces respectively. This is because the vocabulary to express complex movements or relationships between multiple objects is very limited.

We consider that our work is divided into two fields. The first field is, as mentioned before, video databases. The second is moving object databases related to spatio-temporal databases. Other researchers are engaging in moving object databases that support the spatial objects with continuously changing position (Forlizzi et al., 2000) (Nabil et al., 1997). Forlizzi et al. define a discrete data model that is described as a collection of data types and operations which can be plugged in as attribute types into any DBMS data model to obtain a complete model and query language. This model also implements the abstract model of the work by (Güting et al., 2000).

Next, we compare the work by (Erwig et al., 1999)(Erwig and Schneider, 2000) which is similar to ours. They also have the same motivation of formulating queries by the visual specification of movements. They proposed a visual interface in order to specify predicates to be used in queries on spatio-temporal databases. The intersections of a trace with another object's trace is translated into a sequence of predicates, which can represent, for example, *disjoint, meet, equal, inside*, and so on. These predicates are mainly based on the variety of events between moving objects, and they are only interested in specifying topological relationships and need only information about relative positions of objects with respect to each other. In our work, however, we need information about absolute positions of objects for retrieval. This information can be obtained due to recent advances in positioning systems. It should be noted that we have different ways of specifying spatio-temporal relationships of multiple moving objects: they put emphasis on the predicates to represent abstract relationships and not discrete positional relationships between objects by translating visual notation into a sequence of predicates. We emphasize the specification of the discrete temporally changing positional relationships by using our proposed method (see Section 4).

## 3. QUERYING INDIVIDUAL MOVING OBJECT'S MOVEMENT

The snapshot of our prototype is shown in Figure 2. We chose soccer as a subject because it represents video data that captures multiple moving objects. We explain our proposed approach using this prototype. The main component of this prototype is the *canvas* on which the user

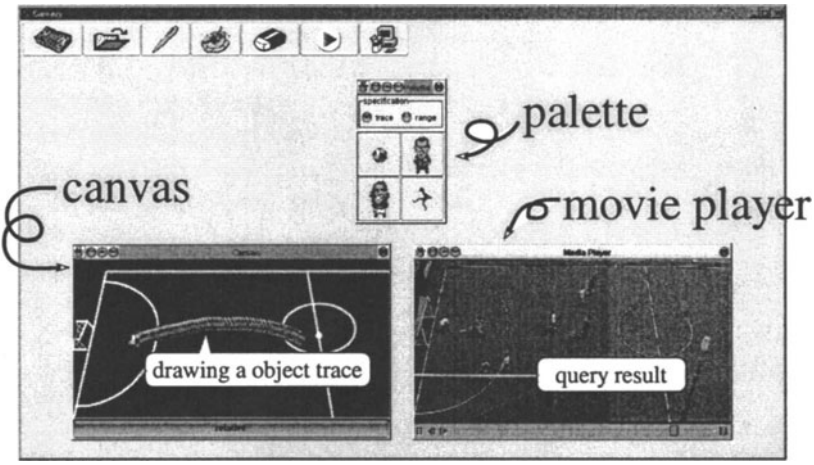


*Figure 2* Snapshot of our prototype

draws a moving object trace. The prototype has a palette by which a user can specify what is the moving object of interest.

We explain a way to query by drawing a moving object trace. Figure 2 shows that by drawing a trace, a user can search for video intervals that capture object movements similar to the drawn trace query. With our prototype, the query process is described as follows:

1) On the palette, the user selects an object of his or her interest.

2) On the *canvas*, the user draws a trace of the specified object.

3) The user pushes a button to retrieve the desired video intervals.

Figure 2 shows that a ball trace drawn on the *canvas* is used as the query, and the query result is played by the movie player.

In order to retrieve video intervals, conventional time-series matching is done between the time-series data of the moving object's positions and the data of the drawn trace on the *canvas*. We use the Dynamic Time Warping (DTW) method (Berndt and Clifford, 1996)(Yi et.al., 1998). DTW is suitable for matching voices, audio frequencies and medical signals. This method can compute the distance between two time-series even if the lengths of the two series are different. Figure 3(a) shows how DTW works.

We explain in detail how a set of desired intervals can be acquired by using DTW to apply two-dimensional time-series data. $S$ is the time-series data given by the drawn trace, and $T$ is the moving object's

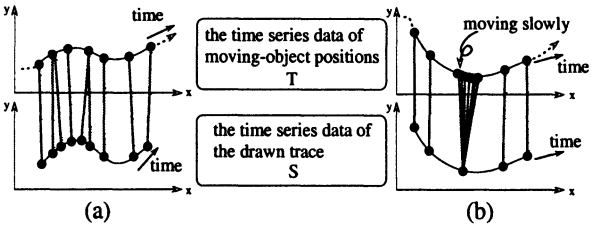


*Figure 3* Dynamic Time Warping

position. Assuming there are two discrete sets of time-series data, $S$ and $T$, which are given as follows: $S = s_0 s_1 \ldots s_i \ldots s_{I-1}$ and $T = t_0 t_1 \ldots t_j \ldots t_{J-1}$, where $s_i$ and $t_j$ are the $i^{th}$ and $j^{th}$ factors of $S$ and $T$ respectively, and $I$ and $J$ are the lengths of $S$ and $T$, respectively. Given the two time-series $S$ and $T$, DTW creates a matching matrix $M[i,j]$, using the difference between the two sets, $d(i,j)$, by the following formulas.

$$
\begin{aligned}
d(i,j) &= |\, s_i - t_j \,| \\
M[0,j] &= d(0,j) \quad (0 \le j \le J-1) \\
M[i,0] &= d(i,0) + M[i-1,0] \quad (1 \le i \le I-1) \\
M[i,j] &= d(i,j) + min\{M[i-1,j], M[i-1,j-1], M[i,j-1]\} \\
&\qquad (1 \le i \le I-1\ ,\ 1 \le j \le J-1)
\end{aligned}
$$

Next, we define the procedure $Q(T_s, T_e)$ that requires two arguments, $T_s$ and $T_e$, as follows:

(1) We define the following value as $e$:

$$min\ \{\ M[I-1,T_s],\ M[I-1,T_s+1],\ \ldots,\ M[I-1,T_e]\ \}$$

If $e$ is more than a certain threshold, the calculation by function $Q(T_s, T_e)$ stops.

(2) We define a $t_e$ that satisfies the equality between $M[I-1,t_e]$ and $e$. We define a $t_s$ that is calculated by tracing the calculation process from $M[I-1,t_e]$ to $M[0,t_s]$.

(3) An interval that is yielded by the starting time $t_s$ and the ending time $t_e$ is regarded as one of the answers.

(4) We calculate the following two formulas, $Q(T_s, t_s - 1)$ and $Q(t_e + 1, T_e)$. That is, the procedure $Q$ has a recursive structure.

By using the defined procedure, we calculate a set of desired intervals. Calculating $Q(0, J-1)$ recursively makes it possible to retrieve a set of video intervals.

The reason why we use the DTW is that the time axes are different in the two time-series: one is a hand-drawn trace and the other is the moving object's positional data. Although the lengths of the two time-series are different, the DTW makes it possible to compute their pattern distance by normalizing the time-axis. However, a problem arises in querying the velocity and acceleration of the object's movement. This is especially obvious when the object's movement is slow. Since the DTW makes the two time-series datasets correspond in a way that minimizes their pattern distance, a particular point sometimes corresponds to an arbitrary number of points, as shown in Figure 3(b). That is, there is no limitation in the number of points in one time-series data that corresponds to a point in another time-series data. This leads to the problem: we cannot distinguish the speeds of the two time-series datasets. In order to resolve this, we limit the number of points that can correspond to a particular point, which makes it possible to protect the time-axis from extreme expansion and contraction.

In this way, we can distinguish fast movements from slower movements *to some extent*. The reason why we say *to some extent* is that the described algorithm is insufficient in treating the object's speed and acceleration. It is difficult to set a threshold value for the number of points that can correspond to a particular point because drawing speed (that is, the length of the two time-series) generally differs from one person to another if more than one person tries to make the same query. This problem will be the subject of future work, at which time we will consider a way to dynamically set the threshold value described above.

# 4. SPECIFYING THE SPATIO-TEMPORAL RELATIONSHIPS OF MULTIPLE MOVING OBJECTS

## 4.1. TIMELINE

In order to formulate a query by drawing traces of the movements of multiple objects, it is necessary to be able to specify the spatio-temporal relationships of these objects. In order to make this specification, we propose the usage of a *timeline*. Figure 4 shows an example for specifying the spatio-temporal relationships between two traces:

1) On the *canvas*, the user draws traces of multiple object (See Figure 4(1)).

2) On the *timeline*, the user specifies the intended relationships by dragging the rows of the *timeline* while watching the bottom row (See Figure 4(2)).

The bottom row of the *timeline* represents the spatial relationships of multiple moving objects in a specific temporal relationship.

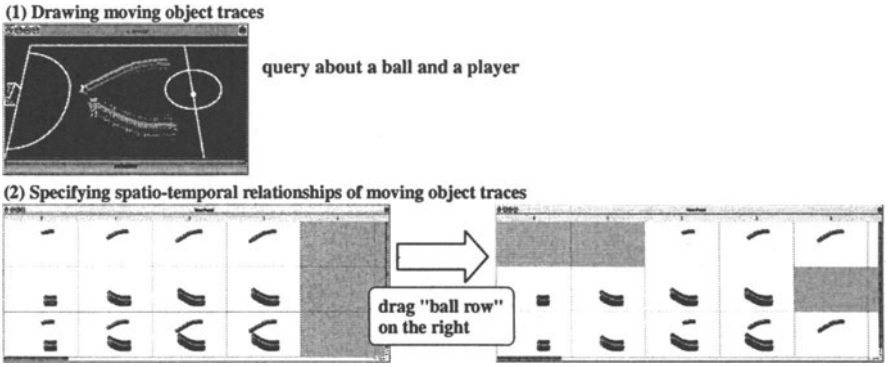


*Figure 4* Specifying spatio-temporal relationships of multiple moving objects by using a timeline

## 4.2. SPATIALLY RELATIVE QUERIES AND MULTI-CANVAS

**4.2.1 Spatially Relative Queries.** We propose a spatially relative query. A spatially relative query means a query to retrieve object movements whose *shape* is similar to a given trace but independent of the position of the trace. The other method, described in Section 3, is called a spatially absolute query. The difference between the two is whether or not the position of the trace is considered. (See Figure 5).

The retrieval process is described as follows: (1) Find the minimum rectangle that includes the drawn trace. This rectangle is known as the template. (2) Employ the "template matching" method. As shown in Figure 6, this is the raster scanning of the given template from the edge of the field. (3) At each point while scanning, the DTW is applied.

The above method, however, requires a very large number of calculations because the whole field is scanned. In order to modify this method, we employ the following technique: if template matching shows a low level of similarity in a certain local domain, the nearby domain is also given a low level of similarity. By employing this technique, any domain

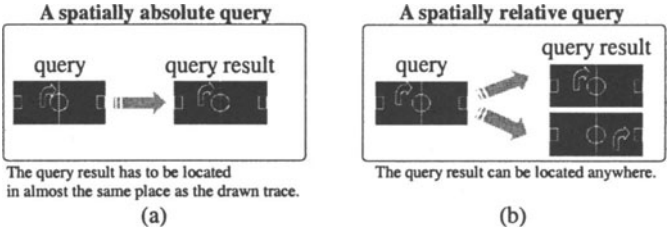


*Figure 5* A spatially (a) absolute and (b) relative query

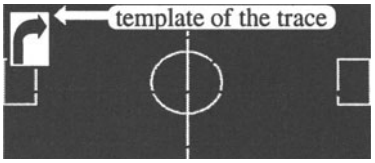


*Figure 6* Template matching

that is expected to have the low similarity according to the DTW can be trimmed, and the comparative number of calculations can be reduced.

**4.2.2 Multi-Canvas.** Let us apply a spatially relative query to multiple moving objects. In order to formulate such a query, it is necessary to provide a user interface that can easily specify spatial and temporal relationships. To do this, we introduce the *multi-canvas*, which can specify the spatio-temporal relationships of multiple moving objects by using multiple *canvases* that are spatially distributed. Figure 7 shows the spatio-temporal relationships of multiple moving objects. Our prototype provides a button that the user can select to make a new *canvas* appear. The user can then freely drag each *canvas* to change its position. By changing the positions of the *canvases* either vertically or horizontally within a *multi-canvas*, it is possible to change a query without having to redraw the traces.

The query shown by Figure 7(a) can be represented by one *canvas*. This query can be interpreted as two traces that are temporally synchronous and spatially absolute (that is, the position of the trace is considered), and produces the same query in Figure 4 but without a *timeline*.

In order to specify the spatial relationships of multiple moving objects by using the *multi-canvas*, we propose to place the *canvases* vertically. We emphasize that the *canvas* whose background is undefined gives a

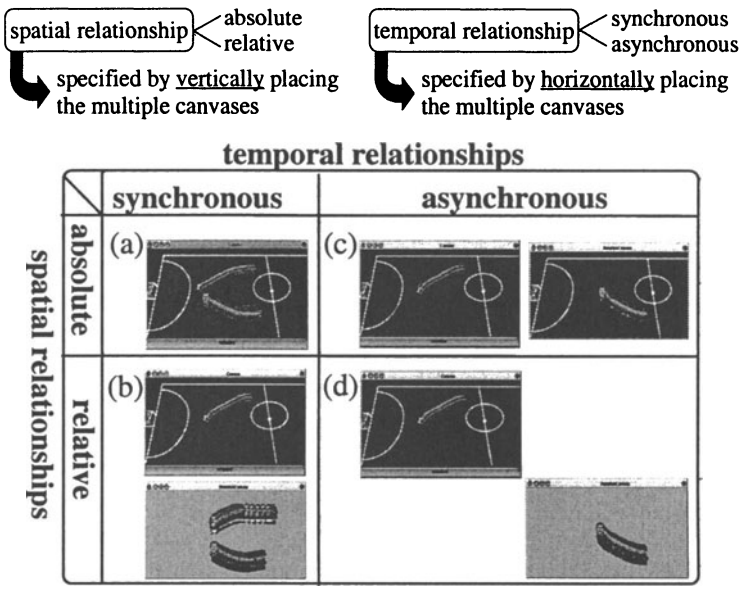


*Figure 7* Specifying spatio-temporal relationships by using a multi-canvas

spatially relative query. The query shown by Figure 7(b) represents a spatially absolute query for the ball in the upper *canvas* and a spatially relative query for the two players in the lower *canvas*. The interval given by the movements of the ball and the two players has to be synchronous. There are spatially absolute relationships between the two players because they are drawn on the same *canvas*. If the two players do not have spatially absolute relationships between them, another *canvas* has to be used. That is, three *canvases* have to be placed in one vertical line.

In order to specify the temporal relationships of multiple moving objects by using the *multi-canvas*, the horizontal axis of *canvases* is used. That is, the horizontal positions of the *canvases* by nature signify the temporal relationship of video. The *canvas* placed on the right corresponds to the video interval that takes place after the interval of the *canvas* placed on the left. If the *canvases* are not placed horizontally such as in Figure 7(a) and (b), the interval given by each object's movement has to be synchronous. However, *canvases* can be placed horizontally in the number of asynchronous relationships.

Figure 7(c) and (d) both show the following movement: a ball moves first, then a player moves. The difference between Figure 7(c) and (d) is whether a spatially relative query is given for the player. The focus here is that, by dragging only the canvas containing the player in Figure

7(c) below, the query will have the exact same meaning as the query in Figure 7(d). Also, the *canvas* whose background is undefined gives a spatially relative query. Dragging the *canvas* changes the background automatically.

## 4.3. COMPARISON BETWEEN THE TIMELINE AND THE MULTI-CANVAS

In this subsection, we compare the *timeline* mentioned in Section 4.1 with the *multi-canvas* mentioned in Section 4.2. More than one *timeline* is can be used to specify the spatio-temporal relationships of multiple moving objects and to give a spatially relative query; however, the query in this case is a complicated procedure. There are two reason for this and the comparison between the *timeline* and the *multi-canvas*. First, the advantage of a *timeline* is that it can specify the temporal relationships of multiple moving objects precisely and in detail. On the other hand, a query by a *multi-canvas* can specify the temporal relationships only as synchronous or asynchronous. In other words, the *timeline* regards time as contiguous instants, and the *multi-canvas* regards time as roughly contiguous intervals. Thus, the *timeline* is preferred when a detailed specification of temporal relationships is desired. Second, given a spatially relative query, more than one *timeline* has to be used, and the query becomes complicated because of the detailed specification. On the other hand, a query by a *multi-canvas* cannot specify spatio-temporal relationships in detail, but it is easier to handle. Thus, the *multi-canvas* is preferred when a simple query formulation is desired.

# 5. QUERY BY DIRECTLY DRAWING ON A VIDEO SCREEN

Television watching is changing because of the rapid advances in digital broadcasting, which makes it possible to watch television in both real time and also by watching past videos in storage. Moreover, we will be able to watch television in a non-linear way, which will make it possible to acquire more detailed or related information while watching. In this section, we propose a way of querying video data by directly drawing on a video screen while watching the video.

As shown in Figure 8, we can consider the following example in a soccer video game. A certain player, receives the pass, gets a shot at the goal, but fails to score. While watching this or any other soccer program, we often think of what would have happened **if** the player had passed the ball to a nearby player instead. At that time a query that considers

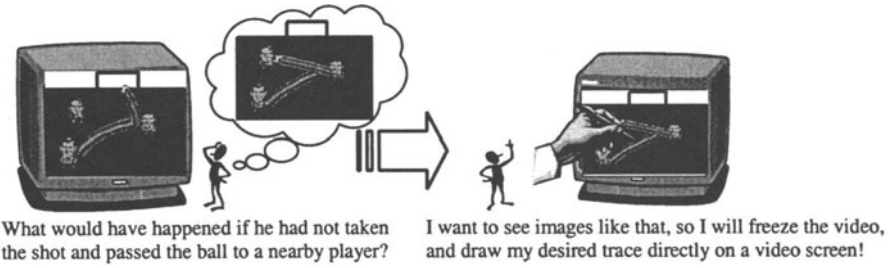


*Figure 8* Concept of a query by directly drawing on a video screen

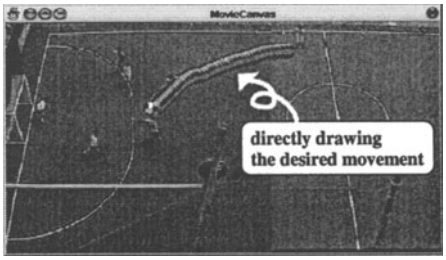


*Figure 9* Query by directly drawing on a video screen

a different movement can be generated. The answer to this query has to be a past video scene that is similar to the current situation.

As shown in Figure 9, our query formulation process is described as follows:

1) The moment a desired query is issued while watching a video, the video is frozen.

2) The user draws the desired movement on the still image.

3) The user presses a button to retrieve the desired video intervals.

The formulated query requires two types of information for processing: (1) the drawn trace on the still image, and (2) the positions of the moving objects at the moment the video is frozen. The positions of all moving objects can be acquired at the specified moment because our targeted video data has the time-series data, which indicates the positions of the moving objects as meta data. Also, when drawing a trace, the user clicks the captured object on the still image; consequently, the system

can specify which object is selected without using the palette mentioned in Figure 2.

The algorithm of retrieving video intervals is described as follows:

1) We use $m$ to denote the number of all objects appearing in the still image and $t_f$ to denote the time the video is frozen. For objects $o_1, o_2, \ldots, o_m$, let us denote the set of their positions at time $t$ by $P(t) = \{p_1(t), p_2(t), \ldots, p_m(t)\}$.

2) For a specified (drawn) trace, retrieve a set of (past) video scenes, each of which contains the movement of a specified object that is similar to the trace. Let us denote the set of starting times of each answer video scene by $T = \{t_1, t_2, \ldots, t_n\}$.

3) Given two sequences, $P(t_f)$ and $P(t_k)$, where $t_k \in T$, the Euclidean distance between the two is calculated for each $t_k$. The smaller the Euclidean distance, the bigger the similarity value between times $t_f$ and $t_k$. If the calculated distance value is less than a certain threshold value, the interval which has the starting time at $t_k$ become the answer (See Figure 10).

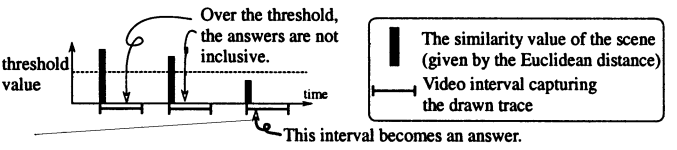


*Figure 10* Retrieving video intervals by directly drawing on a video screen

Next, we consider a method to modify the query result in an incremental manner. That is, we propose to combine the query on the *canvas* and the query on the still image in order to obtain more precise answers in a incremental way. As shown in Figure 11, the query on the still

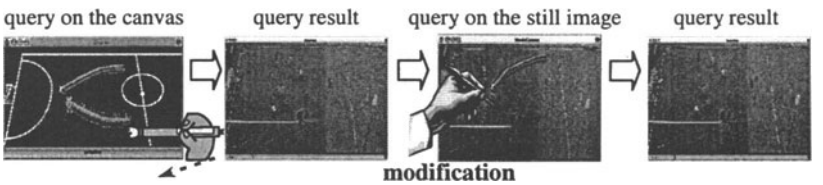


*Figure 11* Modification of the query result

image is used to modify the query result from a previous query on the *canvas*.

This method has two advantages. First, as shown in Figure 11, modification of the query result allows the user to acquire a more precise set of video intervals. Second, the user gives a query on the *canvas* in an abstract manner. Then, adding the modification on the still image of the answer video allows the user to be able to give a more definite query.

# 6. CONCLUSION

We have presented a user interface to query video data that captures multiple moving objects by hand-drawn traces. Using the interface developed in our research allows the user to query video intervals intuitively and easily by specifying the spatio-temporal relationships of multiple moving objects. In order to acquire a qualitative evaluation of our proposal, we asked several people to test our developed prototype after we taught them how to use it. Their summarized comments include that it is an intuitive and easy way of query formulation and it is possible for end users to issue a query for moving objects without having a specialized knowledge.

## Acknowledgments

This research is partly supported by the research for the grant of Scientific Research (12680416 and 13224054) form Ministry of Education, Science, Sports and Culture of Japan.

## References

J.F.Allen. (1983) "Maintaining Knowledge about Temporal Intervals", *Communications of the ACM, 26(11):832-843, 1983.*

E. Oomoto and K. Tanaka. (1993) "OVID: Design and Implementation of a Video-Object Database System", *IEEE Transactions on Knowledge and Data Engineering, 5(4):551-563, August, 1993.*

E.J. Hwang and Y.S. Subrahmanian. (1996) "Querying Video Libraries", *Journal of Visual Communications and Image Representation, 7(1):44-60, March, 1996.*

H. Wactlar, T. Kanade, M. Smith, and S. Stevens. (1996) "Inteligent Access to Digital Video: Informedia Project", *IEEE Computer, 29(5):46-52.*

L. Forlizzi, R.H. Güting, E. Nardelli, and M. Schneider. (2000) "A Data Model and Data Structure for Moving Objects Databases", *SIGMOD Record, 29(2):319-330,June 2000.*

M. Nabil, A.H.H. Ngu, and J. Shepherd. (1997) "Modeling Moving Objects in Multimedia Database", *The 5th Conference on Database Systems for Advanced Applications, Melbourne, Australia, 1997.*

M. Erwig, R.H. Güting, M. Schneider, and M. Vazirgiannis. (1999) "Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases", *GeoInformatica, 3(3):269-296, 1999.*

R.H. Güting, M.H. Böhlen, M. Erwig, C.S. Jensen, N.A. Lorentzos, M.Schneider, and M. Vazirgiannis. (2000) "A Foundation for Representing and Querying Moving Objects", *ACM Transaction on Database Systems, 25(1), 2000.*

M. Erwig and M. Schneider. (2000) "Query-By-Trace: Visual Predicate Specification in Spatio-Temporal Databases", *The 5th IFIP Conference on Visual Database Systems (VDB5), 2000.*

D.J.Berndt and J.Clifford. (1996) "Finding Patterns in Time Series: A Dynamic Programming Approach", *Advances in Knowledge Discovery and Data Mining, pp.229-248, 1996.*

B. Yi, H.V. Jagadish, and C. Faloulsos. (1998) "Efficient Retrieval of Similar Time Sequences Under Time Warping", *The 14th International Conference on Data Engineering (ICDE'98), pp.201-208, IEEE Computer Society Press, Feb. 1998.*

## Biographies

**Chikashi Yajima** is currently a graduate student of the Department of Computer and Systems Engineering at Kobe University. He received his B.E. in Computer Science from Kobe University in 2001. His research interests include multimedia databases.

**Yoshihiro Nakanishi** is currently a graduate student of the Department of Computer and Systems Engineering at Kobe University. He received his B.E. in Computer Science from Kobe University in 2000. His research interests include multimedia databases.

**Katsumi Tanaka** received his B.E., M.E., and Ph.D. degrees in Information Science from Kyoto University in 1974, 1976, and 1981, respectively. In 1986, he joined the Department of Instrumentation Engineering at Kobe University as an associate professor. In 1994, he became a professor of the Department of Computer & Systems Engineering and the Division of Intelligence Science (Information and Media Sciences, later), Graduate School of Science & Technology at Kobe University. Since 2001, he has been a professor of the Department of Social Informatics, Graduate School of Informatics at Kyoto University. His interests include object-oriented databases, historical database models, hypermedia systems, and multimedia information systems. Dr. Tanaka is a member of the ACM, IEEE Computer Society, and the Information Processing Society of Japan (IPSJ).