Models for Motion-Based Video Indexing and Retrieval

Serhan Dağtaş, Wasfi Al-Khatib, *Member, IEEE*, Arif Ghafoor, *Fellow, IEEE*, and Rangasami L. Kashyap, *Fellow, IEEE*

***Abstract*—With the rapid proliferation of multimedia applications that require video data management, it is becoming more desirable to provide proper video data indexing techniques capable of representing the rich semantics in video data. In real-time applications, the need for efficient query processing is another reason for the use of such techniques. We present models that use the object motion information in order to characterize the events to allow subsequent retrieval. Algorithms for different spatiotemporal search cases in terms of spatial and temporal translation and scale invariance have been developed using various signal and image processing techniques. We have developed a prototype video search engine, *PICTURESQUE* (pictorial information and content transformation unified retrieval engine for spatiotemporal queries) to verify the proposed methods. Development of such technology will enable true multimedia search engines that will enable indexing and searching of the digital video data based on its true content.**

***Index Terms*—Content-based retrieval, imprecise querying, motion modeling, video databases, video indexing.**

I. Introduction

RECENT initiatives in digital video technology have significant applications in many areas, including digital libraries, video surveillance, law enforcement, automatic target recognition, traffic management, command and control, etc. The research community in this area is exploring better ways of retrieving information from large repositories of multimedia data. With exponential growth in multimedia data archives, the need to organize, store, search, and display multimedia data has increased tremendously. There is an increasing need for robust indexing and search mechanisms to enable effective use of multimedia data, and in particular, digital video.

Traditionally, the *content* of video has been represented either by simple textual techniques or low-level image features such as color indices, shape representation, transform domain features and visual summary information based on segmentation of video into smaller units. Various scene change detection techniques have been employed to automatically segment video data into shots. A color histogram comparison routine is used in [11] to parse video data into scenes. In [14], an *a priori* model of *reference frames* is used to semantically classify the identified shots into the constituent components of a news broadcasting program. An icon-based browsing environment constructed from the recognized shots is presented in [17]. A hierarchical video stream model which uses a template or histogram matching technique to identify scene changes in a video segment is proposed in [15]. These methods, however, are useful in specific domains, and therefore not readily applicable in the development of general-purpose video data indexing and retrieval systems. In addition, these methods have limited capability since semantics associated with scene changes are captured, but temporal events within a scene are not modeled.

A number of researchers have proposed techniques for video content modeling involving temporal events. Some of these techniques rely on modeling the interplay among physical objects in time along with spatial relationships between these objects. In [6], spatial and temporal attributes of objects and persons are modeled through a directed graph model. Although a formal method of representing video data is proposed, the underlying spatial models for motion-based characterization are not provided, rendering severe limitations to the system. An approach that uses spatial relations for representing video semantics is spatiotemporal logic [2], [5]. In [2], a prototype image sequence retrieval system is developed, where video frames are processed and simple events are represented by spatiotemporal logic. The prototype provides a query interface by which query-by-sketch is employed to query video data. However, spatial and temporal predicates are manually annotated in the database, thus requiring considerable manual effort for event specification. The framework discussed in [8] defines a set of algebraic operators to allow spatiotemporal modeling and provide video editing capabilities. After extracting trajectory of a macro-block in an MPEG video, all trajectories of macro-blocks of objects are time-averaged and a spatiotemporal hierarchy is established for representing video. This technique does not address the handling of interobject relationships and content description due to its limited analysis at lower-levels.

VideoQ [3] is one of the very few models that directly address motion-based content characterization. However, the approach lacks an explicit temporal formalism and comprehensive spatial search techniques that can handle different preferences such as spatial-translation and spatial-scale invariant retrieval.

As mentioned earlier, low level image features are not sufficient to represent the rich semantics of video. Spatiotemporal characteristics involving the relative movements of salient objects in a video must be incorporated in any effective content-based video indexing scheme. This is mainly due to the fact that humans often describe the semantic content in terms of the

Manuscript received December 15, 1998; revised July 27, 1999. The associate editor coordinating the review of this manuscript and approving it for publication was D. A. Murat Tekalp.

S. Dağtaş is with Philips Research, Briarcliff Manor, NY 10510 USA (e-mail: serhan.dagtas@philips.com).

W. Al-Khatib, A. Ghafoor, and R. L. Kashyap are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 479907 USA (e-mail: wasfi@ecn.purdue.edu; ghafoor@ecn.purdue.edu; kashyap@ecn.purdue.edu).

Publisher Item Identifier S 1057-7149(00)00180-9.

action present in video data. *“Clips with a touch-down event”* is a much more meaningful description than *“Clips with green background”*. Similarly, queries like *“search for a head-on collision involving a car and a van”* cannot be answered by indexing techniques based on color histograms. An effective indexing scheme cannot be restricted to preassigned keywords or just be simple raw data that carries no semantic meaning. Powerful intermediate spatiotemporal models are needed that provide enough semantic power while supporting an efficient retrieval mechanism for effective content-based access.

In this paper, we propose several motion description schemes that serve as intermediate spatiotemporal models for event-based retrieval of video. We summarize the basic components of our approach and the major contributions as follows.

- Description of characteristics and identification of requirements for motion-based video retrieval. Based on this discussion, we propose two alternative and complementary schemes: *trajectory* and *trail-based models* for motion-based indexing of video.
- Based on these models, we propose efficient searching algorithms. Specifically, these techniques address the invariance features in spatial and temporal domains.
- For the search case that does not require any invariance feature, we propose computationally efficient searching techniques based on common statistical methods. These methods also provide flexibility to the user in determining the right search parameters for optimum accuracy/performance tradeoff.
- We have developed the *pictorial information and content transformation unified retrieval engine for spatiotemporal queries* (PICTURESQUE), a video database retrieval system that provides an effective example-based querying mechanism and alleviates the limitations of keyword-based search techniques. Our implementation covers the preprocessing of the raw video for subsequent retrieval as well.

The remainder of the paper is organized as follows: In the next section, we outline the characteristics and requirements of motion-based video modeling and present the formal statement of the problem addressed in this paper. Based on this foundation, we present the *trajectory* (Section III) and *trail* models (Section IV). Section V lays out the implementation details and experimentation results of the proposed models. Section VI concludes the paper.

II. Motion-Based Modeling in Video Databases

In our earlier work, we have presented a graphical model called *video semantic directed graph* (VSDG) as a representation scheme for temporal specification of video content [5]. VSDG provides an effective means for interobject temporal specification for multiple objects that enables temporal content based access to the semantics in digital video. Based on this foundation, we now present new models that extend the functionality of VSDG in terms of trajectory specification support that enables object movements to be modeled and queried.

As a graphical data structure, VSDG offers a visual environment where the temporal relations can be effectively represented and constructed. For a more complete representation scheme, however, mechanisms must be defined for an effective spatiotemporal characterization. The mechanism of searching for events in such a system will involve modeling and comparisons of motion features of the salient objects in video. In the next section, we analyze the desired characteristics of such schemes which lead to the two motion models in subsequent sections.

A. Desired Characteristics of Motion-Based Modeling and Retrieval

In order to process user-sketched queries like *give me the clips where an object draws a circle like this (a sketched path)*, a similarity measuring mechanism between motion representations must be developed. Our experience shows that in order to answer such queries effectively, any similarity measurement mechanism must possess certain features. The nature of query processing in a multimedia database system is significantly different from traditional databases in this regard. According to our observations, the major issues pertaining to the video database retrieval include *fuzziness*, *efficiency*, and *space and time invariance*:

- Different user perceptions introduce fuzziness in querying, since a rough sketch may not exactly represent the desired scenario. Furthermore, typically more than one clip is expected to be returned for a given query and these clips will not possess the exact motion features of the query. For these reasons, the similarity criterion should provide a fuzzy measure based on a numeric scale rather than merely accepting/rejecting database items.
- Computational efficiency is an important and often underestimated factor in multimedia database applications. With unrestricted length of the database clips, it becomes more important for a motion similarity mechanism to function with an acceptable complexity for real-time and scalable query processing.
- The issue of spatial invariance arises when the queries may specify a translation-invariant motion which may be located in arbitrary locations in the spatial domain(screen space). However, in some cases it may be desirable to restrict the location to exact coordinates in the spatial domain, such as surveillance applications where the camera is fixed and the environment is defined. Therefore, spatial translation invariance should be optional rather than a default behavior. In addition, spatial scale invariance may be desired for retrieval which is independent of the motion or object dimensions.

  Similar to the spatial invariance, temporal scale invariance refers to the flexibility of the speeds of the object movements. Users may describe motion with arbitrary speeds and may be interested in the general path the objects follow rather than exact locations at exact time instances. For this reason, the similarity matching mechanism should be able to match motion descriptions regardless of the speeds of the objects when necessary.

TABLE I
MOTION-BASED VIDEO RETRIEVAL MATRIX

	AA	**AI**	**IA**	**II**
A	III-A	C	III-B	C
I	IV-A	X	IV-B	IV-C

Similarly, there should be a mechanism to externally manipulate the general duration of the queried events for the users to have control over the temporal features of the described scenario. Translation invariance in temporal domain refers to searching of queried motion throughout the entire temporal domain and is the default behavior.

The first two criteria, regarding the flexibility and performance of the comparison techniques, are subjective features. The invariance properties, however, can be characterized more objectively and associated retrieval techniques can be developed based on such distinctions. In such a framework, depending on the application and the environment, users should have the ability to choose for invariance options in spatial and temporal domains as part of the querying process. These features must be facilitated by the right combination of motion representation and comparison techniques. In other words, the representation scheme is as important as the retrieval technique employed in possessing the right features.

Ideally, a motion representation scheme should possess the features listed above as closely as possible. Among the various motion representation schemes, four are considered the most prominent [7]: B-spline, chain code, differential chain code, and raw trajectory. For majority of the retrieval cases discussed in the next sections, we use the last one, the *trajectory model* due to its flexibility in employing various numerical retrieval techniques. For temporal scale (speed) invariant retrieval, however, this model does not prove to be viable and we propose a new scheme, the *trail model*. This model is based on trail images constructed by highlighting the areas covered by moving objects and is elaborated in Section IV.

Based on the invariance properties discussed above and the two motion representation schemes, we propose several spatiotemporal retrieval methods summarized in the *motion-based video retrieval matrix* in Table I. In this table, we identify six different cases of retrieval methods based on the distinction along temporal scale invariance, spatial translation invariance and spatial scale invariance. The horizontal axes correspond to temporal scale invariant (**I**) and absolute (**A**) and the vertical axes are for combinations of spatial translation and scale invariant cases, respectively. For example, the column **IA** corresponds to translation invariant and scale-absolute case in spatial domain. We denote cases for which a method is proposed here with corresponding section numbers. We do not address three cases in this paper: temporal absolute, spatial scale invariant cases (both labeled with a "C") and temporal invariant, spatial translation absolute-scale invariant case (labeled "X"). For the cases labeled with a "C," differential chain code scheme appears to be a viable approach [8], but is not elaborated here. For the third case (X), we do not propose a solution, however, it can easily be derived from the all-invariant case (**I-II**) as a special case. All others are labeled by their corresponding section numbers.

B. Video Content Organization and Query Formulation

We have emphasized the important role of spatiotemporal characterization of video data for content-based retrieval. We perceive such characterization with an event-centered viewpoint and rather call it *event-based* retrieval. An event can simply be defined as "an interesting happening" in a clip. The formal description of events, however, must be made through a well-defined spatiotemporal characterization model. We construct this description based on the two aforementioned models. We begin this by presenting an overall data organization and then define the query formulation which is the formal problem statement for both models.

In our approach, video data is organized as follows: Raw video is segmented into *clips* that form the atomic unit in the database. Each clip contains several semantic *objects* (cars, humans, etc.) which carry two types of information: *Descriptive_data* and *Motion_data*. Descriptive_data refers to object features like the identity of an object, its color, shape, types, etc., which is not addressed in this paper as part of the proposed motion based retrieval methods. For event-based characterization, spatiotemporal features of moving objects (*Motion_data*) are more important and therefore are the focus of our work. We adopt the minimum bounding rectangles (MBR's) to represent the objects. Despite their limitations, MBR's provide a concise and simple low level representation of the object boundaries.

Formally, *object* = {*Descriptive_data*, *Motion_data*}, where *Descriptive_data* = [*ObjectID*, *Size*, *Color*, $\cdots$], and *Motion_data* = $[C(k), W(k), H(k)]$, which contains $C(k) = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix}$, $k = 1, \cdots, N$, the center point locations of MBR's for the frames that range from 1 to N, and the widths (W) and the heights (H) of the object. The array C has two elements, one for each coordinate axis, x and y. Note that x denotes the sequence (vector) while $x(k)$ corresponds to an individual element. The components of *Motion_data* are used to build the intermediate database indexing scheme for subsequent retrieval of the associated video clip.

In the trajectory model, the center point coordinates, designated by the sequence C, are captured at frame instances (for each k) and form the basis for trajectory models. In other words, the x and y coordinates of the MBR centers define the trajectories of objects on the screen and are used for similarity measurement between query (C^Q) and data (C^D) indices. The matching process for such trajectories can be formally expressed as follows.

For every stored database item C^D in *Database* if $Dissimilarity(C^D, C^Q) < Threshold$ then accept C^D where $C^Q(k) = \begin{bmatrix} q_x(k) \\ q_y(k) \end{bmatrix}$ and $C^D(i) = \begin{bmatrix} x(i) \\ y(i) \end{bmatrix}$ for $k = 1, \cdots, N_q$ and $i = 1, \cdots, N$, N_q is the number of frames in the query sequence and q_x and q_y designate the center-point trajectories for x and y axes.

In the next section, we propose several efficient methods for computation of $Dissimilarity(C^D, C^Q)$ and provide detailed analyses of those methods.

III. Motion-Based Video Retrieval with Trajectory Model

In order to retrieve videos based on their motion characteristics, a similarity measuring mechanism has to be developed that will search the database using the appropriate indexing scheme. In this section, we propose several techniques based on the trajectory model and analyze their effectiveness for several invariance situations. These techniques consist of: 1) temporal scale-absolute retrieval (using the trajectory model), which includes: a) translation-absolute retrieval and b) translation-invariant retrieval; 2) temporal scale-invariant retrieval (using the trail model), which includes: a) spatial translation and scale-absolute retrieval, b) spatial translation-invariant and scale-absolute retrieval, and c) spatial translation and scale-invariant retrieval models.

In this section, we elaborate on case 1), the temporal scale absolute retrieval based on the trajectory model. Trail model [case 2)] is covered in Section IV.

A. Spatial Translation-Absolute Retrieval

Though users will typically require spatial translation-invariant retrieval, exact locations may be an important part of the query in certain situations. For example, in security surveillance video taken with a fixed camera, the location of the moving objects on the screen will be of importance to detect certain events (e.g., illegal right turns) and query processing must be performed to allow such differentiation.

As a common similarity measure, we use Euclidean distance for spatial absolute comparisons (matching) of sequences. This yields the formulation of the $Dissimilarity(Dis)$ and $Distance(Dist)$ functions as

$$Dis(C^D, C^Q) = \min_i [Dist(x_q^i, q_x) + Dist(y_q^i, q_y)] \quad (1)$$

$$\begin{aligned} Dist(x_q^i, q_x) &= (x_q^i - q_x)^T (x_q^i - q_x) \\ &= x_q^{i^T} x_q^i - 2x_q^{i^T} q_x + q_x^T q_x \quad (2) \\ Dist(y_q^i, q_y) &= (y_q^i - q_y)^T (y_q^i - q_y) \\ &= y_q^{i^T} y_q^i - 2y_q^{i^T} q_y + q_y^T q_y \quad (3) \end{aligned}$$

where q_x and q_y are the query sequences, x_q^i and y_q^i indicate portions of x and y with length equal to the length of q_x and q_y and shifted by i, i.e., $x_q^i(k) = x(k+i)$, $k = 1, \cdots, N_q$; $i = 0, \cdots, N - N_q - 1$; $x, y \in Z^N$; $q_x, q_y, x_q, y_q \in Z^{N_q}$; $N_q \leq N$. In order to avoid unnecessary repetition, discussion here on will be done only for X axis. Distances obtained independently are added to compute the minimized $Dissimilarity$ metric in (1).

For translation-absolute retrieval, we carry out the Euclidean distance computation in a translation-absolute fashion. When performed straightforwardly, computation of (2) has an $O(N_q^2)$ time complexity where N_q is the length of q_x, which is in the same order with N. We demonstrate that better performance can be achieved: 1) by computing the Euclidean distances more efficiently or 2) by avoiding unnecessary comparisons. First, we propose an $O(N \log N)$ solution based on computing the three terms in (2) separately and using Fourier transform. A second algorithm we propose computes the distances selectively with a two-stage scheme that eliminates unlikely candidates in the first stage. This makes it possible to achieve a performance even better than $O(N \log N)$ in most cases. Note that this discussion is not limited in scope to trajectory retrieval but also applies to general subsequence matching problems.

1) A Fourier Transform-Based Similarity Computation: Note that the third term $q_x^T q_x$ in (2) is constant for all values of i and can be safely omitted for comparison purposes. The second term involves multiplication of x_q^i and q_x and can be viewed as convolution over the entire x, since x_q^I is a subsequence of x. The elements of the convolution vector are then substituted for the $x_q^i q_x$ multiplication for each x_q^i. It is a well-known rule that convolution in the signal domain corresponds to multiplication in the Fourier transform domain. Therefore, the convolution vector $Conv(i) = x_q^{i\,T} q_x$ can be expressed as

$$Conv(i) = x * q_x^* = \mathcal{F}^{-1}(X \cdot Q_X^*) \quad (4)$$

where X and Q_X^* are Fourier transform of x and q_x^* (the complex conjugated—and padded with zeros for matching dimension—version of q_x), respectively. The advantage of using Fourier transforms to compute the $x_q^{i\,T} q_x$ terms is that Fourier transform can be computed in $O(N \log N)$ time thanks to efficient fast Fourier transform (FFT) algorithms and provides a logarithmic reduction of the computation time of the overall algorithm. The overall complexity is bounded by the complexity of the Fourier transformation step due to the possible linear time computation of the first term $x_q^{i\,T} x_q^i$: elements of the squared term vector $SQ(i) = x_q^{i^T} x_q^i$ can be recursively computed as

$$\begin{aligned} SQ(0) &= x(1)^2 + \cdots + x(N_q)^2 \\ SQ(i) &= SQ(i-1) + x(N_q + i)^2 - x(i)^2 \\ &\quad \text{for } i = 1, \cdots, N - N_q. \quad (5) \end{aligned}$$

As a sequential computation, the above equation results in a linear time complexity. Algorithm RETRIEVAL_ABS, shown on the next page, summarizes the basic steps of the discussed method. Steps 1 and 3 are performed in linear, $O(N)$ time and the Fourier transform in Step 2 is $O(N \log N)$. The overall complexity of the algorithm therefore is $O(N \log N)$.

2) Two-Stage Method: While the Fourier-based method reduces the complexity significantly, further reduction can be achieved by eliminating unnecessary computation of Euclidean distances for each x_q^i. For this purpose, we propose a method that filters out the unlikely candidates by a simpler measure in the first stage and compute the actual distance in the second stage for the items satisfying the first criterion. A natural candidate for the first criterion is the absolute difference between averages of the elements of x_q^i and q_x. Namely, the condition

$$\frac{|Ave(x_q^i) - Ave(q_x)|}{N_q} \leq \tau_1 \quad (6)$$

must be satisfied in order for the Euclidean distance ($Dist$) to be computed for that particular x_q^i. Division by N_q is to allow the right hand side of the inequality (τ_1) to be an independent

```
Algorithm RETRIEVAL_ABS: Fourier-Based Temporal
and Spatial-Absolute Retrieval
Input: Data and Query sequences, $C^D$ and $C^Q$
Output: Dissimilarity between $C^D$ and $C^Q$
Step 1 Construct the complex conjugate of $q_x$
  for $i = 1$ to $N_q$
    $q_x^*(i) = q_x(N_q + 1 - i)$
  for $i = N_q + 1$ to $N$
    $q_x^*(i) = 0$
Step 2 Take Fourier transforms of both signals
  $X = \mathcal{F}(x)$, $Q_X^* = \mathcal{F}(q_x^*)$
Step 3 Using (4) and (5), compute
  $Dist(x_q^i, q_x) = SQ(i) - 2Conv(i)$
  for each $i$, $0 \le i \le N - N_q$
Step 4 Repeat Steps 1–3 to compute $Dist(y_q^i, q_y)$
Step 5 Compute $Dis(C^D, C^Q)$ as
  $\min_i[Dist(x_q^i, q_x) + Dist(y_q^i, q_y)]$
```

variable. A match is eventually decided according to the condition

$$Dist(x_q^i, q_x) = \frac{(x_q^i - q_x)^T(x_q^i - q_x)}{N_q} = \frac{\|(x_q^i - q_x)\|_2^2}{N_q} \le \tau_2. \qquad (7)$$

Main steps of the method are outlined in Algorithm TWO_STAGE, shown at the top of the page.

The philosophy behind this method is that if a sequence is not close to another sequence by its average, it is unlikely that the Euclidean distance will be close enough to grant a match. In the next section, we provide a theoretical analysis of the above formulation and present an optimization case for determining the right parameters for optimal computational and functional efficiency.

3) Analysis of the Two-Stage Algorithm: The correlation between the two criteria, namely the difference of the first and the second norms of the differences, makes it possible to carry out a statistical analysis of the error for the two stage algorithm. The error is defined as the ratio of data items rejected in the first stage that satisfies the second criterion(false negatives). In this section, we provide an analysis of this error and discuss the determination of the optimal values for the thresholds τ_1 and τ_2.

First, we rewrite (6) as

$$\frac{|\,\|x_q^i\| - \|q_x\|\,|}{N_q} \le \tau_1. \qquad (8)$$

Clearly, there is a trade-off between the values of the thresholds, τ_1 in (8) and τ_2 in (7). Their choice determines the "tightness" of the similarity measures in both stages, therefore is important in the resulting error rate. It becomes imperative to express the "success" measures in terms of these parameters. In order to define an error bound for the "misclassification" in the first stage, we define $Error\_Probability(\tau_1, \tau_2)$ as:

```
Algorithm TWO_STAGE: Dissimilarity Computation
in Two Stages
Input: Data and Query sequences, $C^D$ and $C^Q$
Output: Dissimilarity between $C^D$ and $C^Q$
Step 1 For each $x_q^i$, $0 \le i \le N - N_q$
  if $\frac{|Ave(x_q^i) - Ave(q_x)|}{N_q} \le \tau_1$ then
    $Dist(x_q^i, q_x) = (x_q^i - q_x)^T(x_q^i - q_x)/N_q$
  else
    $Dist(x_q^i, q_x) = \infty$
Step 2 Repeat Step 1 to compute $Dist(y_q^i, q_y)$
Step 3 Compute $Dis(C^D, C^Q)$ as
  $\min_i[Dist(x_q^i, q_x) + Dist(y_q^i, q_y)]$
```

$$Prob\left\{\frac{\|(x_q^i - q_x)\|_2^2}{N_q} \le \tau_2 \,\middle|\, \frac{|\,\|x_q^i\| - \|q_x\|\,|}{N_q} > \tau_1\right\}. \qquad (9)$$

To simplify the notation, let $d_k = x_q^i(k) - q_x(k)$. Then the expression in (7) can be written as

$$\frac{\|(x_q^i - q_x)\|_2^2}{N_q} = \frac{d_1^2 + d_2^2 + \cdots + d_{N_q}^2}{N_q}. \qquad (10)$$

Note that the d_k values represent the differences between coordinate values of the query location and the location of the data objects. In order to come up with a statistical bound for the above probability, we assign a random variable for the difference between consecutive d_k values and define them recursively as $d_{k+1} = d_k + R$ where R is a random variable. This will result in the closed form expression

$$d_k = d_1 + R_k, \qquad R_k = r_1 + r_2 + \cdots + r_k. \qquad (11)$$

Therefore

$$\frac{d_1^2 + (d_1 + R_1)^2 + \cdots + (d_1 + R_{N_q-1})^2}{N_q} = \frac{N_q d_1^2 + 2d_1 \sum_{k=1}^{N_q-1} R_k + \sum_{k=1}^{N_q-1} R_k^2}{N_q}. \qquad (12)$$

Lemma 1: Let $x_q^i, q_x \in Z_q^N$ and $d_k = x_q^i(k) - q_x(k) = d_1 + R_k$, $1 \le k \le N_q$ and R_k is a random variable. Given

$$\frac{|\,\|x_q^i\| - \|q_x\|\,|}{N_q} > \tau_1$$

the probability

$$Prob\left\{\frac{\|(x_q^i - q_x)\|_2^2}{N_q} \le \tau_2\right\}$$

has an upper bound equal to

$$Prob\left\{\frac{\sum_{k=1}^{N_q-1} R_k^2}{N_q} - \frac{\left(\sum_{k=1}^{N_q-1} R_k\right)^2}{N_q^2} \leq \tau_2 - \tau_1^2\right\}. \tag{13}$$

Proof: See [4] for a proof.

The fact that the above limit is independent of the initial difference value d_1 is interesting but natural; both criteria deal with the differences of two sequences, and the absolute initial value should not have any effect on the behavior or distribution. The threshold values, on the other hand (τ_1 and τ_2) directly affect the error bound. As expected, the higher τ_1 values decrease the error bound, as a looser first stage limit would reduce the probability of falsely eliminating data items in the first stage. This, however, would have a negative effect on the overall computational complexity of the algorithm which is the main reason such a scheme is used. Therefore, the choice of the thresholds becomes an optimization problem which can be formulated using an error function defined in the following theorem.

Theorem 1: Let $x_q^i, q_x \in Z_q^N$ and $d_k = x_q^i(k) - q_x(k) = d_1 + R_k$, $1 \leq k \leq N_q$ and R_k be a random variable. Define three random variables A, B, C, and D as

$$\begin{aligned} A &= \tau_1^2 - \frac{\left(\sum_{k=1}^{N_q-1} R_k\right)^2}{N_q^2} \\ B &= d_1^2 + \frac{2d_1 \sum_{k=1}^{N_q-1} R_k}{N_q} \\ C &= \tau_2 - \frac{\sum_{k=1}^{N_q-1} R_k^2}{N_q} \\ D &= d_1^2 + \frac{2d_1 \sum_{k=1}^{N_q-1} R_k}{N_q} + \frac{\left(\sum_{k=1}^{N_q-1} R_k\right)^2}{N_q^2}. \end{aligned} \tag{14}$$

Then $Error\_Probability(\tau_1, \tau_2)$ of (9) equals

$$Error\_Probability(\tau_1, \tau_2) = \frac{Prob\{A < B \leq C\}}{Prob\{D > \tau_1^2\}}. \tag{15}$$

Proof: See [4] for a proof.

In order to determine a distribution and a possible optimum threshold value, we need the distribution of R_k in (11). This will then lead to the computation of the probability in (15) as a function of the thresholds which can be optimized.

Using the corresponding probability distribution and cumulative distribution functions, one can express the conditional probability of (15), which is a function of the variables τ_1 and τ_2 as

$$\int_{x \in \mathcal{R}(A,\, B,\, C)} \frac{f_B(x) F_{A|B}(x|x)[1 - F_{C|B}(x|x)]\, dx}{[1 - F_D(\tau_1^2)]} \tag{16}$$

where $\mathcal{R}(A, B, C)$ is the intersection of the ranges of the random variables A, B, and C.

Clearly, the distributions for A, B, C, and D are needed in order to compute the above probability. If the distribution function of R_k's is known, these functions can be determined by analytical or numerical methods. Recall that $R_k = r_1 + r_2 + \cdots + r_k$ and $r_j = (x_q(j+1) - x_q(j)) - (q_x(j+1) - q_x(j))$, i.e., the difference of the differences. One possibility is to assume that r_j (the differences of coordinate values in consecutive frames) has a normal distribution with zero mean as often done in many applications. In this case, the random variables R_k will also be normal with zero mean by the well known rule that sum of normal variables is also normal [12]. The same rule also applies to the distribution of the term $\sum_{k=1}^{N_q-1} R_k$, as the summation would be another normal random variable with zero mean and higher variance. Similarly, the distribution of R_k^2 can be assumed exponential although it would have a discontinuity at zero. This simplifies the analytical form of F_C as the sum of gamma distribution [10].

The normal distribution assumption for the object movements in consecutive frames of the video data will reduce the complexity of the computation of the error probability of the two-stage algorithm. However, numerical simulation and functional approximation is still needed for the computation of the integral in (16). This will also allow arbitrary distributions for the actual data to be correctly processed, removing a restrictive assumption.

As the last step in formulating the optimization problem, we express the inverse of what Theorem 1 formulates: the probability (ratio) of the data items accepted in the first stage and eliminated in the second, i.e., the *false alarm* rate. This ratio is critical in the overall efficiency of the algorithm, because the unnecessary computation of the Euclidean distances in the second stage will significantly increase the complexity and eliminate the advantage of the two-stage scheme with respect to the Fourier-based method discussed in the previous section.

As an application of the α and β error optimization concept in statistics, we define $False\_Alarm(\tau_1, \tau_2)$ as

$$Prob\left\{\frac{|\,\|x_q^i\| - \|q_x\|\,|}{N_q} \leq \tau_1 \,\middle|\, \frac{\|(x_q^i - q_x)\|_2^2}{N_q} > \tau_2\right\}. \tag{17}$$

Following the same line of argument, we can express the above probability as

$$\frac{Prob\{C < B \leq A\}}{Prob\{E > \tau_2\}} \tag{18}$$

which can alternatively be expressed [similar to (16)] as

$$\int_{x \in \mathcal{R}(A,\, B,\, C)} \frac{f_B(x) F_{C|B}(x|x)[1 - F_{A|B}(x|x)]\, dx}{[1 - F_E(\tau_2)]} \tag{19}$$

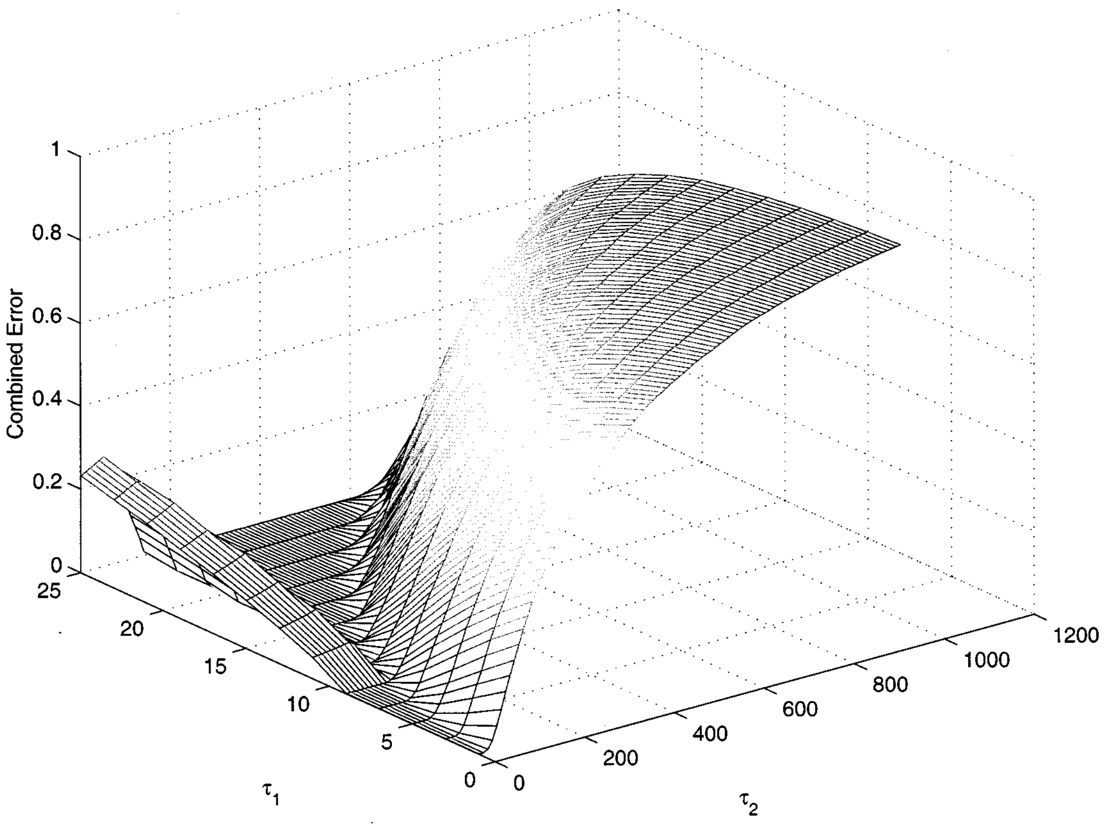


Fig. 1. Combined error plot for different values of τ_1 and τ_2.

where the variable E is defined as

$$E = d_1^2 + \frac{2d_1 \sum_{k=1}^{N_q-1} R_k}{N_q} + \frac{\sum_{k=1}^{N_q-1} R_k^2}{N_q^2}$$

and F_E is the corresponding cumulative distribution function.

Using the functions in (16) and (19) we express the optimization problem as

$$\min_{\tau_1, \tau_2} \beta_1 Error\_Probability(\tau_1, \tau_2) + \beta_2 False\_Alarm(\tau_1, \tau_2)$$
$$\text{subject to } \tau_1, \tau_2 \geq 0 \qquad (20)$$

where the elements *Error_Probability* and *False_Alarm* are as defined in (16) and (19), respectively.

Now, we demonstrate a numerical illustration of the above nonlinear optimization problem and how the optimum values for the thresholds can be computed. An important factor is the choice of the coefficients β_1 and β_2. As in any optimization problem, their values have a deterministic effect on the optimum threshold values. Conceptually, β_1 and β_2 are the weights assigned to the relative importance of the error rate and compromise in the computation efficiency, respectively. While β_2, the weight of the redundant computations, can easily be quantified in terms of the ratio of the complexity of the second and first stages, β_1, the cost of the error is not as easy to describe quantitatively. Its choice, therefore, is left to the user at the time of the query entry within a certain predetermined range to allow the user to dynamically determine the tradeoff between the accuracy and efficiency.

As an illustration of this analysis, we have carried out a numerical simulation of (20), the plot of which is provided in Fig. 1. In this simulation, we have generated queries with $N_q = 100$. The values of τ_1 range between one and 25, whereas τ_2 goes from one up to 1000. Under the assumption that differences (d_k values) in the distances between every two points in the query have a normal distribution with 0 mean and a variance of two, we have generated 10 000 queries. In this graph, the combined error (objective function in the optimization problem) is plotted for different values of τ_1 and τ_2. The graph contains two nonzero sections; the right hand side of the figure is mainly due to the *Error_Probability* part and the left side originates from the *False_Alarm* part of the objective function. Using such numerical data, one can determine a feasible (optimum) τ_1 value that keeps the error under certain limit and does not compromise computational efficiency with high τ_1 values. This is due to the fact that *False_Alarm* increases with increasing τ_1 values. In this graph, the flat areas where the probabilities are near zero would represent the optimum combinations of the thresholds. These areas will be larger for distributions with small variances due to smaller estimation error, and hence less error probability.

A major advantage of using the proposed two-stage method (in addition to possible performance gain) is the flexibility it offers to determine the right tradeoff between performance and precision. The combination of threshold values in the above formulation determine the compromise between precision

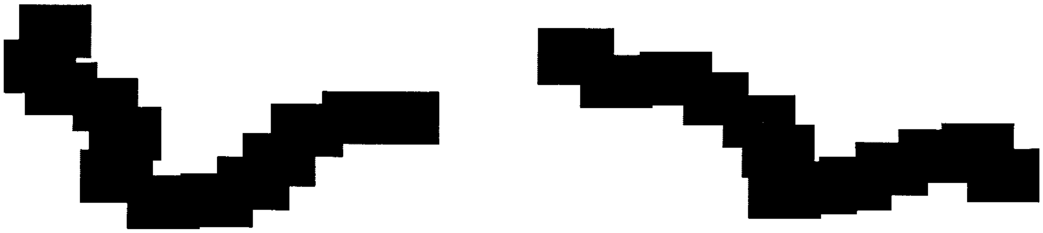


Fig. 2. Example trails used in the trail-based match method.

(*Error_Probability*) and performance (*False_Alarm*). Note that even with optimization, two-stage algorithm will not offer the best solution in all cases. If the precision requirement is too tight (i.e., the elimination in the first stage does not significantly reduce the overall complexity), then the Fourier-based distance computation discussed in the previous section will be superior in terms of the computational complexity. The "breaking point" will be determined by the distribution parameters, in other words, the motion characteristics of the retrieved video.

B. Spatial Translation-Invariant Match

An important factor for robust retrieval of video based on motion is the spatial-invariance, the ability for retrieval of motion trajectories irrespective of their exact locations on the screen. In such a case, a query trajectory would be compared in a translation-invariant fashion against the data trajectories. If the user's choice is to search for the pattern anywhere in the screen, which may often be the case, the plain exact matching algorithms of the previous section will not return the desired results. This is due to the fact that the offset between the two sequences will have an accumulative error in the overall dissimilarity measure and change the results significantly. In this case, Fourier transform-based computation will not work due to its aggregate computation of the entire sequence at once. We therefore perform computations of the Euclidean distance individually for each subsequence after compensating for the initial value, i.e., compute the Euclidean distance plainly at the expense of the lower efficiency. The results for the spatial invariant case are presented in Section V.

IV. Temporal Scale-Invariant Video Retrieval Based on Trail Images

In order to retrieve video clips independent of their temporal characteristics (speeds of the moving objects), we use a trail-based model that captures the motion of salient objects over a sequence of frames. In this method, we highlight the areas covered by the (bounding boxes of the) objects throughout the course of its motion as illustrated in Fig. 2. This results in an "image" of the trajectory for each object. In a sense, this is equivalent to taking the mosaic image of the object trajectory in a clip. The motion comparison is then carried out using the trail images by performing an image similarity comparison that mainly measures the overlap of two trails. For example, the objects in Fig. 2 span similar trajectories and therefore, their resulting trail images have a large overlapping area. Two such images can be compared in different ways including spatial-invariant, spatial-absolute, rotation-invariant, etc.

Trail model is inherently temporal scale invariant due to the fact that the time information is not preserved during the construction of the trail images (for a similar approach and further elaboration on its scale invariance, see [3]). Time is essentially frozen throughout the clip and the varying speed of the object is not reflected in its trail image. However, the duration information of each trail is recorded so that it can be utilized by the higher level queries that may also involve the durations, e.g., *Search for circular motion that lasts between 10 and 20 s*. Such external control capability on temporal duration provides flexibility in user description of the complex events.

An important factor that must be considered in this method is the impact of the temporal length of the clips. When converted to trail images, very long clips will lose their trajectory information as the repeated scans of the same area is not reflected in the binary trail image representation. For the method to be effectively used, this factor has to be taken into account at the time of parsing the video data.

Both spatial-absolute and invariant retrievals can be an option, where the user may choose to either restrict the starting point of the motion or perform a translation-invariant search for the desired motion regardless of the exact locations on screen. A third case is where spatial scale invariance is required. The query trail in this case is searched independent of the size of the object or the dimensions of the trail. For example, a small circle and a big circle can be matched to each other and are considered "similar" in this method. Below, we propose algorithms that efficiently compute the similarity between two trails according to all three cases.

The algorithm Trail_Retrieval, shown at the top of the next page, summarizes the steps of our trail-based retrieval method. According to the user's choice of the retrieval type, the associated images that represent the clip as a motion trail are compared in three different ways. The output of the algorithm, *Similarity* is sorted and the "best N matches" are displayed according to the user's choice of the number N in the user interface implementation.

A. Spatial Translation and Scale-Absolute Retrieval

For a spatial-absolute retrieval, the user inquires for a motion trail that occurs in an absolute screen location. In this case, two trails such as those in Fig. 2 are directly compared against each other for a pixel to pixel match. The fact that the trail images are binary images provides a significant performance advantage, the comparisons are merely a bitwise multiplication between the

```
Algorithm TRAIL_RETRIEVAL: Motion-Based Retrieval using Trail Model
Input: Data objects $O^D = [C^D, W^D, H^D]$ and
  Query object $O^Q = [C^Q, W^Q, H^Q]$
Output: Similarity between $O^D$ & $O^Q$
For each $O^D$ and the given $O^Q$ do
  Construct data & query images $D(i, j), Q(i, j)$
  with 0 for background & 1 for areas covered by trails.
  If User_Choice = Spatial_Absolute_Retrieval
  then $Similarity = sum(D * Q) * (sum(D) + sum(Q))/.5$
  elseif User_Choice = Spatial_Invariant_Retrieval then
    Compute the Fourier transforms of the images as $D_F = \mathcal{F}(D)$ & $Q_F = \mathcal{F}(Q)$
    $Similarity = max(abs(\mathcal{F}^{-1}(D_F * Q_F)))*$
      $(sum(D) + sum(Q))/.5$
  elseif User_Choice = Scale_Invariant_Retrieval then
    Similarity = SCALE_INV_RETRIEVAL $(D, Q)$
```

corresponding pixels. The Similarity step in the algorithm in this case has a quadratic time complexity $O(N^2)$, N being the width or height dimension of the input trail images, which is generally proportional to the screen size.

B. Spatial Translation-Invariant and Scale-Absolute Retrieval

Spatial-invariant retrieval refers to the comparison of two trails in a translation-invariant fashion. This involves the comparison of two images for all possible translations in both dimensions and is computationally intensive. As an efficient way to compare the images in such a fashion, we use the convolution property of the Fourier transform which can be stated as

$$Conv[D, Q] = D * Q = \mathcal{F}^{-1}(D \cdot Q^*). \tag{21}$$

With an FFT implementation of the Fourier transform, this step can be reduced to an $O(N^2 \log N)$ time complexity.

C. Spatial Translation and Scale-Invariant Retrieval

For matching two trails independent of both their starting points and their sizes, we use a Mellin transform-based scale invariant pattern recognition technique which is summarized in Algorithm SCALE_INV_RETRIEVAL [1], shown at the top of the page. This method provides both spatial translation invariance and spatial scale invariance, due to the scale-invariant nature of the Mellin transform and the convolution scheme used in the algorithm.

Mellin transform of a discrete-time signal $x(k)$ is given by

$$[\mathcal{M}(x)](u) = \sum_{k=0}^{N-1} x(k)k^{-(ju+1)}. \tag{22}$$

```
Algorithm SCALE_INV_RETRIEVAL: $(g_1, g_2)$:
Scale-Invariant Retrieval using Trail Images
Input: $g_1(x, y), g_2(x, y)$: $(Z^N \times Z^N) \rightarrow \{0, 1\}$
Output: Similarity between trail images $g_1$ & $g_2$.
1. Calculate the 2-D discrete Fourier transform
   $G_i(f_x, f_y)$ of $g_i(x, y)$
   where $f_x, f_y = -(N/2), \cdots, 0, \cdots, N/2 - 1$, & $i = 1, 2$.
2. Take absolute value of the transform & normalize all values to the maximum value at zero frequency.
   $H_i(f_x, f_y) = \frac{|G_i(f_x, f_y)|}{G_i(0, 0)}, i = 1, 2.$
3. Logarithmically distort $H_i$ in the $f_x$ & $f_y$ direction, putting the result in $D_i, i = 1, 2$.
4. Compute the measure function
   $M(k) = \frac{\left(\sum_{u=k}^{N-1}\sum_{v=k}^{N-1}[D_1(u, v) - D_2(u - k, v - k)]^2\right)^{1/2}}{\left(\sum_{u=k}^{N-1}\sum_{v=k}^{N-1}[D_1(u, v)]^2 + [D_2(u, v)]^2\right)^{1/2}},$
   $k = 0, 1, \cdots, N - 1$
5. Repeat 3 & 4 for the upper left quadrant, or the lower right quadrant.
6. Compute the Similarity by inverting the dissimilarity measure $DSM = \min(M)$.
```

Scale invariance of Mellin transform can be easily proven by substitution. For $x_\alpha(k) = x(\alpha k)$

$$[\mathcal{M}(x_\alpha)](u) = \alpha^{-ju}[\mathcal{M}(x)](u). \tag{23}$$

Therefore

$$|[\mathcal{M}(x_\alpha)](u)| = |[\mathcal{M}(x)](u)|. \tag{24}$$

Another property of the Mellin transform is its close relationship to Fourier transform. Mellin coefficients can be easily computed from Fourier coefficients by scaling the input signal by a logarithmic scale. Substituting $l = \log\, k$ one can show that

$$[\mathcal{M}(x)](u) = [\mathcal{F}(x(e^l))](u) = [\mathcal{F}(x)](\log\, u) \tag{25}$$

which is the basis of Step 3 in the algorithm. For details on this scale-invariant method, please refer to [1].

It is worthwhile to comment on the shift and scale invariance option of the algorithm TRAIL_RETRIEVAL in more detail. Typical user queries will not specify the desired object motion in its exact scale and translation. In other words, it may be desirable to retrieve all object movements resembling a specified trajectory regardless of their exact location on the screen or what the dimensionality of the trajectory is. For example, the query *give me the clips with right to left passes* from a football clip can be answered correctly only if the algorithm can retrieve cases with different scales in both the size of the object and the size of the

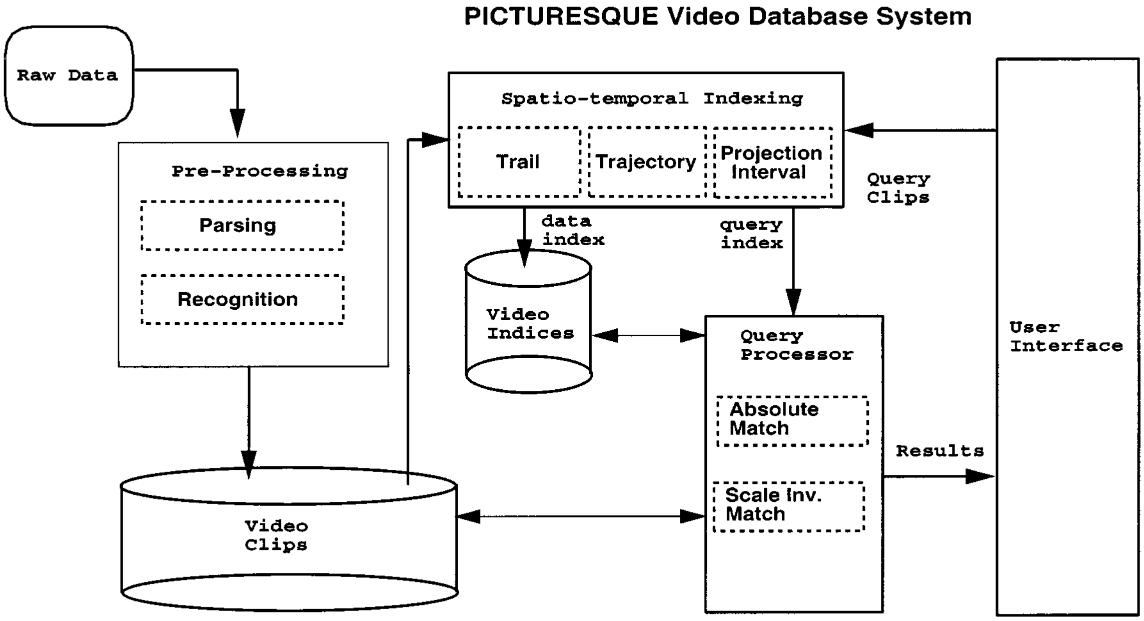


Fig. 3. Overall architecture of PICTURESQUE.

trajectory (e.g., longer or shorter passes). The scale invariant algorithm is used in such cases to retrieve video clips independent of the object and trajectory scales as demonstrated in the next section.

V. Implementation and Experiments

We have built a prototype video indexing and retrieval engine, PICTURESQUE, as a testbed for the methods we have proposed in this paper. The tool was implemented in Windows platform and consists of two main components: Video Motion Indexing Tool (VMIT) and Video Search Engine (VSE).

The architectural components of the system is depicted in Fig. 3. In this framework, VMIT comprises of the preprocessing and spatiotemporal indexing modules and VSE contains the query processor and user interface modules. According to this model, a user can query the video database by first specifying trajectories of objects. Then, based on the mode of retrieval (spatial-absolute, spatial-invariant and scale-invariant), proper algorithms are invoked, and the input trajectory is compared with the data items stored in the database in the query processor module. The retrieved data items are ranked according to their similarity and the corresponding video clips are returned.

In order for the video clips to be accessed by such a system, raw data has to be processed first, which is done in the preprocessing module. Incoming video clips are first indexed, and the spatio-temporal indexing schema is constructed according to three models: trajectory, trail (as discussed in Sections III and IV) and projection intervals (not subject of this paper). This schema is used in the query processor module to search for the desired clips based on their motion characteristics.

During the preprocessing step, objects of interest are identified and their position and size are specified with a bounding box, MBR. Despite their known limitations, MBR's provide an efficient way to represent approximate location of objects on the video coordinate space. Automatic detection and recognition of objects for this purpose is an extremely challenging task. It is widely accepted that with the current state of the art in the technology, software tools can most effectively be used as an aid to human users for the purpose of extraction of the "interesting" information, and fully automated indexing is far from being accomplished. Toward this goal, we use a semi-manual object tracking tool for capturing MBR's where we use limited object tracking based on error thresholding. In this method, the encapsulated objects (areas within the bounding boxes) are tracked as long as a user-set threshold is not exceeded in the error between consecutive frames. An alternative method could be indexing at certain intervals and interpolating for "interframes" to reduce the redundancy at high frame rates.

The requirement of preprocessing of video within the proposed framework is a significant shortcoming and is a general handicap in content-based multimedia access. The widespread use of the upcoming content-aware video representation standards such as MPEG4 or MPEG7 are expected to help alleviate such problems in the future.

The query tool (VSE) is similarly used for retrieval of the data objects according to the constructed indices. In order to enter a query, a rectangle (MBR) is drawn for each object and dragged on the screen to specify the motion trajectory to be searched. During this process, its coordinates are recorded at specified time intervals in real time. Prerecorded object trajectories can be played back while the new trajectories are entered thereby allowing a multiobject metaphor.

A. Experiments with the Trajectory Model

We have used the PICTURESQUE tool to measure the performance of the models and algorithms we have proposed in this paper. In doing that, we have faced the common challenge in the multimedia research area: Measuring retrieval performance of video objectively is very difficult given the complexity of the

TABLE II
Trajectory Model Retrieval Results: Translation-Absolute Queries

Clip Name	Abs. Rank	Over Ave.	Over Second	Inv. Rank	Over Ave.	Over Second
Balloon	1	.32	.56	3	.15	X
Car	1	.14	.07	1	.11	.04
Train	1	.03	.14	2	.10	X
Dancer	1	.04	.33	1	.16	.48
Fish	1	.19	.81	5	.38	X

TABLE III
Trajectory Model Retrieval Results: Translation—Invariant Queries

Clip Name	Inv. Rank	Over Ave.	Over Second	Abs. Rank	Over Ave.	Over Second
Balloon	1	.31	.58	13	3.41	X
Car	1	.04	.10	8	.91	X
Train	3	.27	X	13	3.24	X
Dancer	1	.12	.39	10	1.42	X
Fish	3	.38	X	1	.15	.45

data and subjectivity of the "success" criteria. There is no analytical or concrete way of measuring the quality of the results similar to peak signal-to-noise ratio (PSNR) technique in traditional signal processing. In addition, due to relatively short history of the video databases, there is no commonly used data for benchmarking the retrieval methods. For these reasons, we have to rely on home-grown methods to report the results until a universal performance measuring framework emerges in the video database field.

In order to partially overcome these shortcomings we have chosen to use a common data set and performed testing with the MPEG7 sample sequences that are distributed as part of the experimentation effort for the upcoming standard. These clips are accompanied with their object segmentation information which allows practical object indexing. In this set, there are a total of 13 sequences with an average length of approximately 12 s. We have extracted the center point locations from these and supplied to our trajectory-based retrieval mechanism.

Due to the limited number of items in the data set and the aforementioned reasons, traditional recall-precision experiments cannot be used for a conclusive testing. We have therefore devised the following technique for performance evaluation of our trajectory algorithm: we have picked five distinctive sequences from the data set and asked the user to query for each one. Then, the similarity measures between the query sequence and 13 data items are computed and ranked. The motion types associated with each of these sequences are as follows: balloon: bouncing; car: horizontal; train, zigzags; dancer, circular; fish, horizontal (right-to-left followed by left-to-right).

Table II shows the results for absolute translation queries (where the user inquires the position as well as the trajectory) and Table III lists the results with translation-invariant queries (where a given trajectory is searched in the entire search space). In these tables, the first two columns contain the results for the intended case (Absolute or Invariant) and the other case is also shown as a reference. "Abs. Rank" indicates the rank of the desired sequence in the results list (1 being the best and 13 being the worst). "Over Ave." is the ratio of the dissimilarity of the desired sequence to the average of all, which signifies the overall differentiability of the metric with smaller numbers indicating a better measure. "Over Second" is the ratio to the second (for the first ranks), an indication of how distinctive the "right pick" is.

As the tables indicate, trajectory model produces generally satisfactory results with the MPEG7 data set. Translation-absolute queries result in more accuracy as the employed metric is more strict and the user is given *a priori* information about the location of the motion. The reduced accuracy with translation-invariant scheme using the same query (second half of the same table) is due to the fact that there may be similar motions with a different location and matches the query better than the desired motion. The results in this column are similar to the results for invariant queries in Table III. The reverse, however, is not true because two motions will not be regarded similar unless their locations match in an absolute search mode, as shown

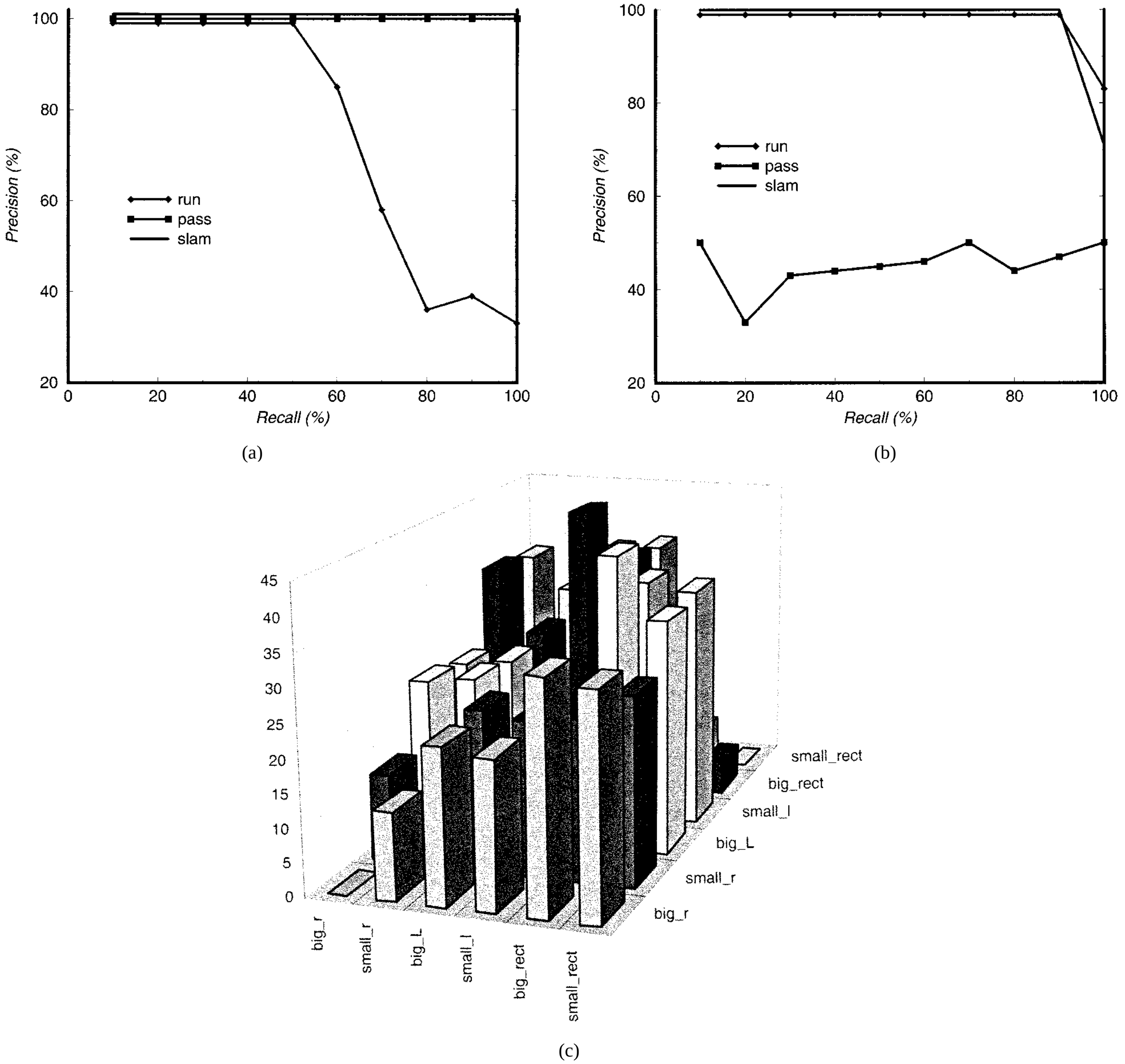


Fig. 4. Recall-precision graphs for the trail model: (a) spatial-absolute retrieval, (b) spatial-invariant retrieval, (c) distance results between sample images for scale-invariant retrieval with SCALE_INV_RETRIEVAL.

in the second portion of the table. An exception is the fish sequence which is merely a coincidence.

B. Experiments with the Trail Model

In order to test the effectiveness of the Trail Model, we have experimented with each retrieval case in the algorithm TRAIL_RETRIEVAL and obtained the results shown in Fig. 4. In this case, we have used a data set that consists of 30 manually generated sequence indices with an average length of 7.5 s. The recall-precision graphs proves to be more useful with a larger data set with more representatives from each ground-truth categories.

For the spatial-absolute and invariant cases our sample data set contains three groups of motions: *run*, *pass*, and *slam*. The run category represents players running from left to right in a football video clip, *pass* represents balls following a parabolic trajectory and *slam* refers to reflection of the ball. Each group contains an equal number (ten) of video clips that are preclassified into the group manually. The length of the sequences ranges between 5 and 10 s. A member from each group is then picked as a query clip and compared against the entire data set. The resulting recall-precision graphs in Fig. 4 indicate that all three algorithms generally produce satisfactory results. In the spatial absolute retrieval, the precision of the *run* query drops rapidly due to the larger size of the associated object (player), as it is easier for other objects to have large overlapping areas with a larger object for other objects. Higher precision for *run* in spatial invariant case in the second diagram indicates the importance of this option for better retrieval of the desired behavior. Low precision of the *pass* query in the same diagram is proof that in some cases a more robust retrieval technique will be needed than a mere translation invariance.

For testing the scale invariant retrieval algorithm, we used a different data set and a more definitive measuring technique.

The data set in this case consists of three user-sketched trails in two different sizes that resemble the letters r, L, and a rectangle, chosen for a better naming convention. The results are remarkably accurate: in each of three categories, dissimilarity measure of Algorithm SCALE_INV_RETRIEVAL (D, Q) gives distinctively close distances between associated classes (small_r to big_r, etc.) In general terms, this retrieval type gives the most natural and expected results, but has severe computational disadvantage. For this reason, it is concluded that the spatial-absolute and spatial-invariant methods should be used as "quick and dirty" searches and the scale-invariant algorithm should be deployed for higher precision retrieval.

C. Discussion and Comments

In this section, we comment on several issues that pertain to the general framework discussed in this paper.

1) Multiple Object Queries: With the trajectory model, scenarios involving multiple objects can be described using pairwise combinations of the objects involved in the same scene. The difference vector define the relative position of objects with respect to each other and is input to the same algorithms as in the case of single object descriptions under the assumption that this vector sufficiently defines the relative movements. This is feasible only when limited number of objects are involved in each scene, as is the case with most real life situations. With the trail model, multiple objects and their relationships have to be handled with an external model for which VSDG is an excellent example. For more detail on this model, please refer to [5].

2) Precision-Performance Tradeoff: In [9], a prefiltering method has been proposed similar to our two-stage algorithm to reduce the computational cost. While the two methods share the same basic idea, *completeness* property enforced in this method restricts the solution space. By allowing inaccuracy, our method offers the ability to adaptively adjust the parameters for the desired precision-performance combination as formulated by the optimization problem in Section II.

3) Temporal Invariance: We have demonstrated that the trail model performs better in cases where retrieval must be temporal scale invariant. VideoQ [3] offers the same feature as "Spatial Mode" where the trajectory is similarly projected onto the x–y space. However, the size in this case is only an external variable, which may be an advantage or disadvantage depending on the retrieval characteristics and our scale-invariant retrieval case partially alleviates the disadvantage of the trail model. In addition, PICTURESQUE addresses spatial translation and scale invariance, and computational efficiency issues explicitly.

4) Object Sizes: Note that object sizes are inherent to the trail model and are treated as an external variable in the trajectory model. The limitations of the trail model in this regard is partially reduced by the scale invariant algorithm. In its current implementation, the size is assumed unchanged throughout the query and this is one of the possible improvement areas of the model. When the trajectory model is used, size is assumed to be an external feature such as the color, identity or other object features and has not been considered in the experiments presented in this paper.

5) Camera Motion: Another important issue is the handling of the camera motion. We compensate for the camera motion, by eliminating the effect of inter-frame camera movements by techniques similar to those used in *salient stills* [16] and *mosaicking* [13]. In these techniques, a still image representation of a clip is obtained by combining several consecutive frames of a clip, a process that can also be used for detecting the movements of the salient objects.

VI. CONCLUSION

We have presented PICTURESQUE, a video indexing and retrieval tool for efficient formulation and processing of user queries based on object motions. The proposed scheme covers many aspects of a video database system from processing of raw video for subsequent indexing to spatiotemporal data modeling. The example-based nature of the visual query tool offers a user-friendly interface as well as a semantic generality and flexibility of the user queries. We have proposed two complementary models for motion-based video characterization that lead to an effective content-based retrieval mechanism for video data.

REFERENCES

[1] J. Altmann and H. Reitbock, "A fast correlation method for scale- and translation-invariant pattern recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 46–57, Jan. 1983.

[2] A. Del Bimbo, E. Vicario, and D. Zingoni, "Symbolic description and visual querying of image sequences using spatio-temporal logic," *IEEE Trans. Knowl. Data Eng.*, vol. 7, pp. 609–622, Aug. 1995.

[3] S.-F. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong, "A fully automatic content-based video search engine supporting multi-object spatio-temporal queries," *IEEE Trans. Circuits Syst.Video Technol.*, vol. 8, pp. 602–615, July 1998.

[4] S. Dagtas, "Spatio-temporal content characterization and retrieval in multimedia databases," Ph.D. dissertation, Purdue Univ., West Lafayette, IN, 1998.

[5] Y. F. Day, S. Dagtas, M. Iino, A. Khokhar, and A. Ghafoor, "Object-oriented conceptual modeling of video data," in *Proc.11th Int. Conf. Data Engineering*, Taipei, Taiwan, R.O.C., Mar. 1995, pp. 401–408.

[6] ——, "Spatio-temporal modeling of video data for on-line object-oriented query processing," in *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, May 1995, pp. 98–105.

[7] N. Dimitrova and F. Golshani, "R_X for semantic video database retrieval," in *Proc.ACM Multimedia'94*, San Francisco, CA, pp. 219–226.

[8] F. Golshani and N. Dimitrova, "Retrieval and delivery of information in multimedia database systems," *Inform. Softw. Technol.*, vol. 36, pp. 235–242, May 1994.

[9] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 729–736, July 1995.

[10] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, New York: McGraw-Hill, 1991.

[11] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full video search for object appearances," in *Proc. 2nd Working Conf.Visual Database Systems*, Oct. 1991, pp. 119–133.

[12] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, Ne York: McGraw-Hill, 1991.

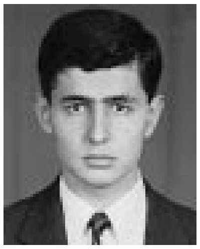
[13] H. S. Sawhney, S. Ayer, and M. Gorkani, "Model-based 2d&3D dominant motion estimation for mosaicking and video representation,", Tech. Rep., IBM Almaden Res. Lab., Dec. 1994.

[14] S. W. Smoliar and H. Zhang, "Content-based video indexing and retrieval," *IEEE Multimedia*, vol. 1, pp. 62–74, Summer 1994.

[15] D. Swanberg, C.-F. Shu, and R. Jain, "Knowledge guided parsing in video databases," in *Proc.SPIE'93*. San Jose, CA, Jan. 1993.

[16] L. Teodosio and W. Bender, "Salient video stills: Content and context preserved," in *Proc. Multimedia'93*, pp. 39–46.

[17] M. M. Yeung, B.-L. Yeo, W. Wolf, and B. Liu, "Video browsing using clustering and scene transitions on compressed sequences," in *Proc. IS&T/SPIE Multimedia Computing and Networking*, 1995.



Serhan Dağtaş received the B.S. degree in electrical engineering from Bilkent University, Turkey, in 1991, and the M.S. and Ph.D. degrees in electrical and computer engineering from Purdue University, West Lafayette, IN, in 1994 and 1998, respectively.

He is currently a Senior Member of Research Staff at Philips Research, Briarcliff Manor, NY. His research interests include multimedia data modeling, image and video retrieval, relational and object-oriented databases, image processing and computer vision.

Dr. Dağtaş was a Fulbright Scholar from 1992 to 1994.

Wasfi Al-Khatib received the B.S. degree in computer science from Kuwait Universityin 1990, and the M.S. degree in computer science from Purdue University, West Lafayette, IN, in 1995. Currently he is a Ph.D. candidate in the School of Electrical and Computer Engineering, Purdue University.

His research interests include multimedia information systems, artificial intelligence, and software engineering.

Mr. Al-Khatif is a member of the ACM, the UPE, and the IEEE Computer Society.

Arif Ghafoor (M'83–SM'89–F'99) received the B.Sc. in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 1976, and the M.S., M.Phil and Ph.D. degrees in electrical engineering from Columbia University, New York, in 1977, 1980, and 1984, respectively.

After graduation, he joined the faculty of the Department of Electrical and Computer Engineering, Syracuse University, Syracuse, NY. In Spring 1991, he joined the faculty of the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, where he is currently a Professor and the Coordinator of Distributed Multimedia Systems Laboratory. This laboratory is a facility for conducting research in multimedia databases, distributed computing, and broadband multimedia communication. He has been actively engaged in research areas related to parallel and distributed computing, and multimedia information systems. He has published more than 130 technical papers in leading journals and conferences. He has recently coedited a book entitled *Multimedia Document Systems in Perspectives* (Boston, MA: Kluwer, 1998). He has been a consultant to GE, the DoD, and the UNDP.

Dr. Ghafoor has been invited frequently to give seminars and tutorials at various leading IEEE and ACM conferences, including keynote speeches at the 13th British National Conference on Databases (1995) and 1997 IEEE Workshop on Resource Management in Computer Systems and Networks. He has served on various IEEE and ACM Conferences Program Committees. Currently, he is serving on the editorial boards of numerous journals including ACM/Springer Multimedia Systems Journal and the Journal of Parallel and Distributed Databases. He has served as a guest co-editor for special issues of numerous journals, including *ACM/Springer Multimedia Systems Journal, Journal of Parallel and Distributed Computing, International Journal on Multimedia Tools and Applications*, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.

Rangasami L. Kashyap (M'70–SM'77–F'80) received the Ph.D. degree from Harvard University, Cambridge, MA, in 1965.

He is currently a Professor of Electrical and Computer Engineering and Associate Director of the National Science Foundation supported Center for Collaborative Manufacturing, Purdue University, West Lafayette, IN, since its inception in 1985. He has been on the faculty at Purdue since 1966, and has directed over 45 doctoral dissertations. He has authored or coauthored more than 400 publications, of which more than 150 are in archival journals.

Dr. Kashyap is the recipient of many honors, including the Kin-sun Fu Award in 1990 for fundamental contributions to pattern recognition and computer vision given by the International Association for Pattern Recognition (IAPR), the J. C. Bose Award, in 1991, for contributions to engineering sciences from the Institute of Electronic and Telecommunication Engineers, election to the status of Fellow of the Institute of Electrical and Electronic Engineers (IEEE), and the Institution of Electronics and Telecommunication Engineering. He is an area editor for the journals *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing,* the *Journal of Intelligent and Robotic Systems*, and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He has been guest coeditor of IEEE TRANSACTIONS ON SOFTWARE ENGINEERING (1988), IEEE COMPUTER MAGAZINE (1989), and the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS (1992). He received the best research paper award at the National Electronics Conference in 1967.