空間時間性視訊資料搜尋之內容表示法

Content Representations for Spatio-temporal Video Data Searching

張 厥 煒
(Chueh-Wei Chang)
Department of Computer Science and Information Management
Providence University,
Taichung Shalu, Taiwan, ROC.
cwchang@pu.edu.tw

李 素 瑛
(Suh-Yin Lee)
Institute of Computer Science and Information Engineering
National Chiao Tung University,
Hsinchu, Taiwan, ROC.
sylee@csie.nctu.edu.tw

ABSTRACT

In this paper, we focus on the problem of design a fast searching method in a video information system to locate video segments that match a content-based query, approximately by time series feature values. The basic idea is to extract video contents via low level feature extraction and/or high level semantic retrieval mechanisms according to a specific point of view, then segment video contents into bounding boxes via a box segmentation mechanism by their time series feature values. Video content indexing is constructed by the characteristics of prominent points that accompany bounding boxes. We also propose an efficient and effective video content matching algorithm to find similar sequences. With the help of the video indexing and matching mechanisms, several high level box-to-box and low level point-to-point query types can be requested.
Keywords: Video Indexing, Content-based Retrieval,.

1. INTRODUCTION

Basically, we can classify content-based video queries into four categories as follows : Type-1 query by alphanumeric data and answer by alphanumeric data; Type-2 query by alphanumeric data and answer by video data; Type-3 query by video data and answer by alphanumeric data; Type-4 query by video data and answer by video data. In Type-3 query, there should contain many kinds of video computing and representations, with high or low level temporal data and spatial data interpretations [7][8]. For Type-1 and Type-3 queries, even though these queries involve accessing video data, the answer is just a list of text strings. Type-2 and Type-4 queries ask for relevant video footage. With formal definition, *content-based retrieval* of video data is a retrieval process based on the understanding of the semantics of the objects in a collection [9]. Content-based video query allows incompletely specified queries, which are processed through a knowledge module [10]. Most early video content retrieval systems are text-based, where relevant text keywords and/or annotations are attached to each video sequence as the basis for retrieval [3][4][11]. Unfortunately, the users of such systems sometimes need to provide a long list of textual query constraints to locate the desired video sequences in the video database. Several researches have been done in content-based retrieval for image and video data [12][13], but they do not provide the capability for content matching in the temporal extension.

In order to manage information in video data, a video information system must be provided. A number of special requirements distinguish the video information system design approach from traditional databases. A video information system needs complex structural representation of its multi-level contents. Video content in a video information system can be represented as a text-type keyword, a paragraph of words, a related image. It allows a user to generate queries containing both temporal and spatial concepts, and also provides content-based searching. However, how to extract and compare video contents in a video information system is still an important problem to be solved.

Therefore, the problem we deal with is the design of a video information system with an efficient video content representation, an effective multi-level query processing capability, and a fast searching method. According our researches, frame-to-frame object changing is one of the most obvious information in video data. With temporal extension, frame-to-frame object changing cause a series of frame-by-frame data. This frame-by-frame time series data is essential to many areas, such as gesture recognition in human-centered information systems, dynamic industrial processed monitoring, scene segmentation [1][2], automatic object tracking, and dynamic scene understanding. That is, the searching method should include an indexing and a matching mechanism that can search a video information system by time-series feature values or even by multi-level semantic meanings, in order to locate video subsequences that match a query sequence approximately. Furthermore, time-series data indexing and matching mechanism can also be applied to many other applications, such as banking, policy decisions, inventory control, and scientific databases, where the history and prediction are important.

In current video database systems, only fundamental techniques, such as keyword-based searching [3], hierarchical video icon browsing and indexing [4], are provided. Most of the previous researches in video data are focused on motion and scene analysis. Very little work has been done on the design of index structures that combine spatial and temporal attributes for video databases.

In this paper, we provide several algorithms to solve these indexing and content-based matching problems. In Section 2, we define the video representation and evaluation model for a video sequence. In Section 3, we show the bounding box concepts, box segmentation and indexing mechanism. In Section 4, we solve the video content approximation matching problem starting from the definition of a similarity measure. After that, a time-series video content query processing mechanism is proposed. Section 5 includes concluding remarks.

2. VIDEO CONTENT REPRESENTATION

2.1. Video Segment Description Model

A video segment is a sequence of video shots concatenated by scene transitions (e.g. fade in/out, cross dissolve, ... etc.) A meaningful scene is a video segment with the result of continuity in perceived, temporal or spatial dimensions from the view points of the users. These temporal and/or spatial meanings in a video segment change frame-by-frame. By using this frame changing information contained in video frames, we can overcome many difficulties, e.g. measuring the speed of a car, encountered in interpreting a single video frame.

In our definition, a video segment is a meaningful scene, $V=\{v_i, v_{i+1}, ..., v_{i+r-1}\}$, where v_i is the starting frame of a video sequence with frame number (or time code) i, and r is the duration of this segment. A video segment consists of several meaningful objects, such as a dog, a color, or even a thought, appearing in this video segment. That is, each video segment has content attributes and associated attribute values to describe the contents. Prior to storing video content into the video information system, the video content annotation module must first identify the relevant objects automatically or manually, then give descriptive representations of objects. Therefore, we design a *Video Segment Description Model* (VSDM) with the annotation structures and related operations [17]. Annotations of a video segment can be described by attributes with several different data types. They can be a text type keyword, a paragraph of words, a related spatial position in one video frame, a series of specific video features in this video sequence (time-series data), or even another content related video segment. In this paper, we only address the indexing and matching problem on time-series data of video segments.

2.2. Time-series Data and Point of View

In this subsection, we explore the relationships between the time-series data and the point of view. From a video segment, the frame changing can occur in combination of primitive feature(s), such as color, size, shape, and/or high level feature(s), such as action and timing, used to describe objects or behavior of objects in the video frames. After the image processing, annotation, or media conversion processes [10], a sequence of raw video data can be transformed into a variety of attribute values of text, or numerical data types with temporal extension. A specific point of view, abbreviated as a *view*, in a video sequence can be represented by a special projection of these features. The evaluation value of a specific view generated by domain knowledge can be a single real number obtained from the combinations of relevant features in a single video frame and this evaluation value is application dependent. For example, the evaluation value can be a weighted sum of relevant features, or some other formula specified by users and/or domain knowledge. In different video applications, different view points and different similarity criteria may be required. Those relevant feature values for the evaluation value can be calculated by image processing or feature extraction routines. Using the notions proposed in [12] and [18], similarity measure of evaluation value can be specified according to different application domains.

2.3. Evaluation Function

For a specific view in a video sequence, we call this frame-by-frame time-series evaluation values the evaluation function of this view, as defined in Definition 1. Each evaluation function can be treated as a function of time, and also be called a *curve* in this paper. Notice that an evaluation function can be a mapping from multi-dimension relevant feature space to one dimension evaluation value, the design of an evaluation function should take care of the problem of similarity ranking.

If we use the surveillance of road traffic as our example of application [5], the average car velocity can be an evaluation function of this application. The overall approach of this application is based on a moving object recognition procedure. A moving object in one video frame is searched for in a succeeding video frame. If the corresponding moving object is found, the velocity is calculated from the positional shift and perspective transformation. That is, the average car velocity is the specific view of this application, and the car velocity is obtained by calculating a combination of several position features extracted from the video sequences as well as the help of specific model knowledge.

DEFINITION 1. *An evaluation function of a video segment V according to a specific view with q features is defined as*

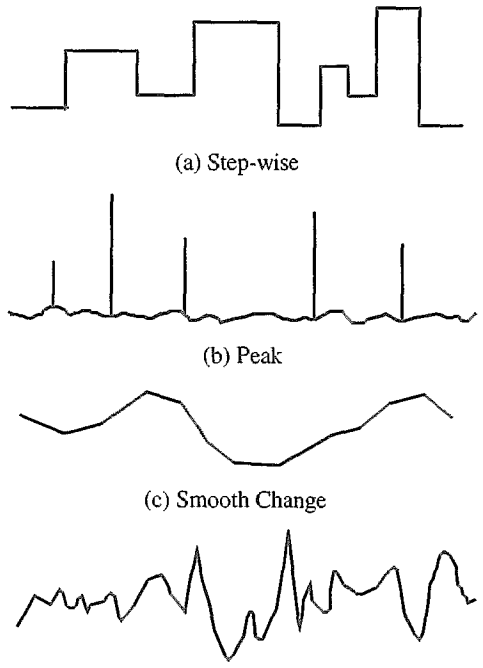
$$E(V, A, T_s, T_e) = \triangle(Ti), \quad (1)$$

where $\triangle$ is the formula of relevant feature vector combination; Ti is the time interval from starting frame T_s to ending frame T_e; $A=[f_1, f_2, ..., f_q]$ is a set of features for the specific video view. We use $E(t)$ that stands for the single evaluation value at time t for a specific video segment and point of view.

3. VIDEO CONTENT SEGMENTATION AND INDEXING

3.1. Curve Distributions of Evaluation Functions

Before we provide a segmentation strategy, we first examine several typical curve distributions which occur in



(a) Step-wise

(b) Peak

(c) Smooth Change

(d) Irregular

Figure 1.Curve distribution in sequence evaluation.

time-series video contents. Figure 1(a) shows the changing of semantic meaning in the video segment, or the variation over time in the number of a certain object (e.g. cars on a street.) We say that this curve distribution has the property of being step-wise constant during each interval. In Figure 1(b), several large peaks appear in this curve distribution (e.g. the frame difference in a cut detection process.) This case is very important since it represents the suddenly happened events. Figure 1(c) shows a situation of smooth changing (e.g. slow motion object or slow color intensity change.) Figure 1(d) represents the randomly distributed irregular curve (e.g. fast action). According to our census, the evaluation functions of video sequences are irregularly distributed in most of the video applications. To overcome this irregular distributed time-series data indexing and matching case, we need to design a feature point finding and segmentation mechanism.

3.2. Bounding Box Principle

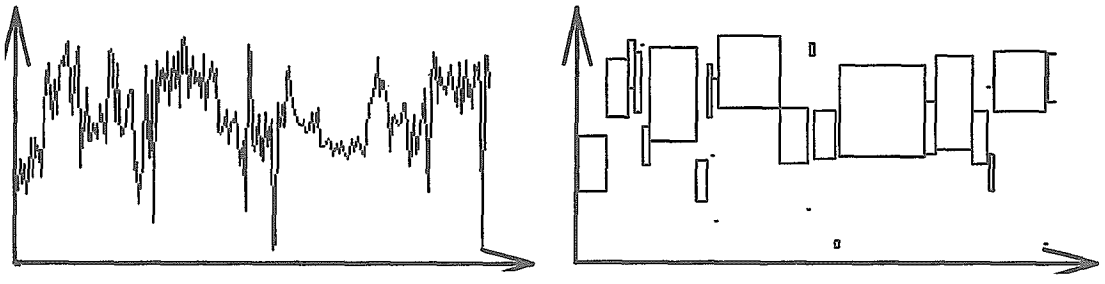
In this paper, we propose a *bounding box principle* as the basis of curve segmentation mechanism. Because contents in video segments can be represented as streams of symbols, the bounding box concept is motivated by the problems which arise in fields of pattern matching and similarity measure. As stated in Section 2.3, we can define an evaluation function that analyzes a video segment by using its low level features, such as representative color, and/or high level semantic meaning, such as the running and jumping of a person. Therefore, each video segment can be segmented into several structured units by a set of special evaluation values or semantic meanings. This mechanism of video segmentation using bounding box principle is so called video structuring, and the result is a structured video.

In consequence of segmentation, the video subsequence between two special successive feature points can be separated and bounded by a bounding box. That is, the evaluation function of a video segment can be divided into a series of bounding boxes by special feature points, as shown in Figure 2. We call these special feature points the prominent index points (or *prominent point*, for short.) Each segmented subsequence is represented as a rectangle box with prominent point value and related information. Except the prominent point value, the following box features can also be included in the related information if necessary : sequence and box ID, minimum and maximum values in this box, offset of duration (box length), inter-box connection type, starting frame/time number, density information of box, previous and next subsequence linkages, high level semantic meaning.

3.3. Box Segmentation and Prominent Points

According to the curve distributions, several kinds of curve features can be found. We classify the curve features into four categories. They are : suddenly up edge, suddenly down edge, increase out of range, and decrease out of range in a curve. We can derive seven connection types from these four categories. Notice that connection type 0 is used for the unstable area, such as a starting and an ending box.

- *Connection Type 1* : large pulse when edge up and down happens in a short time period, e.g. 1/30 second, as shown in Figure 3.



(a) Original sequence

(b) Divided bounding boxes

Figure 2. Transformation between evaluation function and bounding boxes.

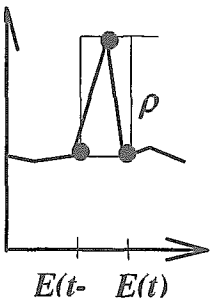
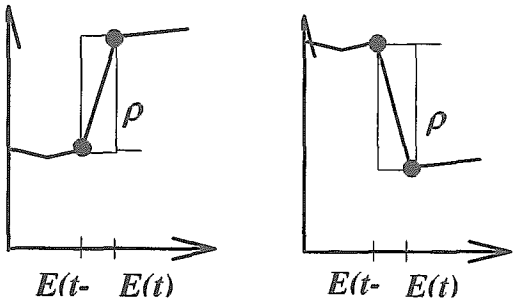


Figure 3. Large pulse case.

- *Connection Type 2, 3* : edge up/down, as shown in Figure 4.

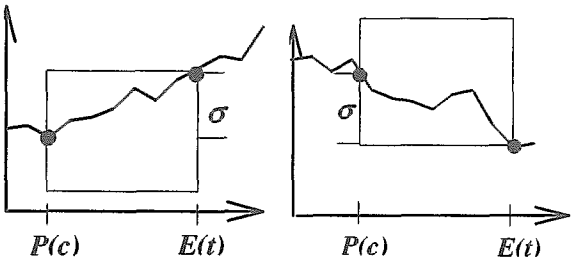


(a)$E(t-E(t-1)>\rho$ for edge up (b)$E(t)-E(t-1)<-\rho$ for edge down

Figure 4. Edge up/down cases.

where *E(t-1)* and *E(t)* are the evaluation function values at time *t-1* and *t*, respectively.

- *Connection Type 4, 5* : increase/decrease, as shown in Figure 5.



(a)$E(t)-P(c)>\sigma$ for increase (b)$E(t)-P(c)<-\sigma$ for decrease

Figure 5. Increase/decrease cases.

where *E(t)* is the evaluation function value at time *t*, *P(c)* is the value of current prominent point.

- *Connection Type 6* : long duration λ of steady situation, as shown in Figure 6.

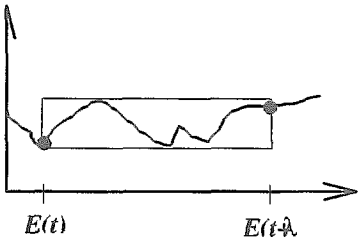


Figure 6. Long duration case.

Therefore, a prominent point can be defined, as shown in Definition 2, by these seven connection types. The parameters ρ, σ, and λ are used to justify whether two sequences are similar. They could be either user-defined, or determined automatically by the distribution of time-series data. The method that we use is to find the prominent points with a large peak of value change (Connection Type 1, 2, 3) or with local increase/decrease in evaluation value in the data stream (Connection Type 4, 5). No matter how the data stream is shifted, the edge-type prominent point of this evaluation function is unique. Same or similar curves will get same or similar prominent points if they follow the same prominent point definitions. That is, if we find two curves with same or similar sequence of prominent points and similar related information, we can say that they are approximate. We can take advantage of this observation of prominent points for efficient indexing in a large database.

DEFINITION 2. *A prominent index point of evaluation function at time t is the point P that satisfies at least one of the following conditions :*

(1) An evaluation value at time/frame t has a change from previous point t-1

$$|E_t - E_{t-1}| > \rho, \quad (2)$$

where E_t is the current evaluation value at time/frame t, and ρ is the threshold value.

(2) The evaluation value difference between the previous prominent point and the current evaluation point is greater than a threshold

$$|E_t - P_c| > \sigma, \quad (3)$$

where E_t is the current evaluation value at time/frame t, and P_c is the current prominent point at time/frame c. σ is the threshold value.

(3) The time/frame difference between the previous prominent point and the current evaluation point is greater than a threshold λ.

3.4. Video Indexing

We use B-tree [19] as our index structure with prominent points as the keys because B-tree has the efficient storage structure and is a robust access method for data points. For each new bounding box, inserting a new prominent point in the index tree is done by a searching index tree and adding the prominent point in a node. The related information about this bounding box is stored in the storage space of a corresponding link list structure and can be accessed through a link list pointer accompanied with the prominent point in the leaf node. Overflowing nodes are split and splits are propagated to parent nodes. If the prominent point in the index tree has already existed, the related information of this new bounding box will be attached at the front of this corresponding link list. That is, the linked list refers to index structure in which an index point may be associated with a list of reference fields pointing to video sequences that contain the same or similar prominent points. By using

the linked list, we can easily find the bounding boxes with similar prominent points and similar box shapes.

4. VIDEO SEQUENCE QUERY PROCESSING

4.1. Multi-level Video Content Query Types

In a video information system, it is necessary to be able to locate some or all occurrences of similar box patterns quickly. We know that the contents of a video segment should be expressed in terms of a set of low level primitive features, and/or combine low level features to form more complex high level semantics. For example, we can specify the query "A person walks on the sidewalk, then suddenly runs across the street and sits on a street chair" by a high level query pattern "(walk)(run)(sit)". Another example is the sequence of video shot types for parsing of news episode in [2]. From the bounding box principle, no matter what kind of video content expressions, the query patterns are considered to be a sequence of values provided by query bounding boxes.

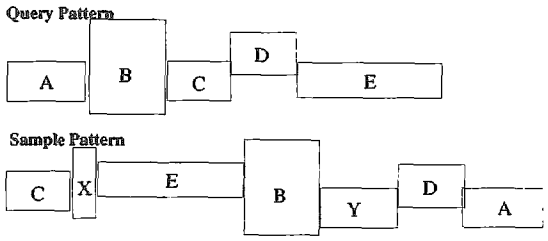
The demand of finding an exact match between two video segments of specific view might be too strict since the real numbers may vary widely. In most of the video applications, users often require finding close or similar but not necessarily exact occurrences. Alternatively, an approximate matching is to find all subsequences in sample video sequences that are close to a query video segment according to some similarity criteria. Therefore, multi-level approximate queries of video segments can be diversified into several categories:

1. Box-to-Box Matching
- Existence Matching : Find those shortest sample box sequences that for each box in the query box sequence, there exists at least one box, which has the same box type, in the matched sample sequence. The order of box types in the matched sample sequences can be neglected. An example is shown in Figure 7(a). Notice that sample and query sequences do not need to have the same box length. The letters in each box stand for the semantic meaning or the prominent points in each bounding box.
- Sequence Matching : Find those shortest sample box sequences that for each box in the query box sequence, the corresponding box in the sample box sequence has the same box type and also has the same order.
  case 1. Exact Sequence Matching - exact one-to-one mapping, as shown in Figure 7(b).
  case 2. Partial Ordering Matching - can have a redundant pattern within sample sequence, as shown in Figure 7(c).

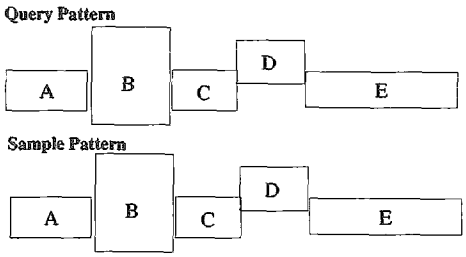
2. Point-to-Point Matching
- Exact Curve Matching : find those sample sequences that the corresponding values are exactly the same as query values.
- Approximate Curve Matching With Error Tolerance : find those sample sequences that the distance between query and sample sequences are within the tolerance of similarity threshold. In other words, those candidates should have a similarity relation for each corresponding value.

Both types of point-to-point curve matching are based on a definition of the *good-match* [20] criterion, as defined in Definition 3. The good-match retrieval is to find the sequence of patterns or evaluation values that are sufficiently similar within some distance ($\tau \sim 0$). With the definition of the good-match, the matching approach should find those patterns or evaluation values close to the search pattern in the video information system within a similarity threshold.

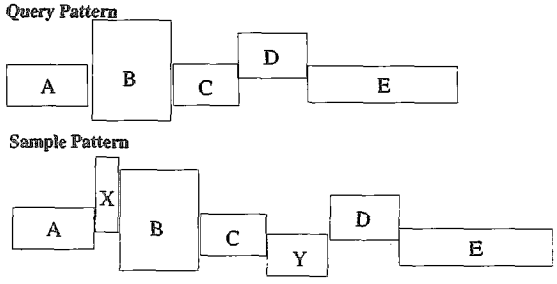
DEFINITION 3. *Given two sequences of patterns, $X = x_1x_2...x_n$ (sample pattern) and $Y = y_1y_2...y_m$ (query pattern), over an infinite alphabet of real numbers, where n and m are respective length of sequence X and Y, if there exists a position (alignment) k in X such that for each pair of corresponding alphabet in these two sequences the similarity measure is smaller than the similarity threshold τ, then subsequence $X' = x_kx_{k+1}...x_{k+m-1}$ is a good-match with Y.*



(a) Existence matching



(b) Exact sequence matching



(c) Partial ordering matching

Figure 7 Query types.

4.2. Query Constraints and Similarity Measure of Query Types

The query result for each query type by some specific query constraint is a set of qualified candidates and the candidates' similarity factor. The result returning from the filtering process of query constraint is a list of qualified candidates that have passed all the checking of the selection conditions. The *similarity factor* (or accumulated penalty) is the summation of similarity measures between the query pattern and the qualified sample patterns for the selected query types.

Except the prominent point, a query constraint can be composed by the selection of the box connection type, min./max. range, box density, box aspect ratio, semantic meaning, etc. For example, the box density is defined as the average value of the accumulated difference between two consecutive evaluation values within the same bounding box,

$$D(B_s, B_e) = \sum_{t=B_s+1}^{B_e} \frac{E(t) - E(t-1)}{(B_e - B_s - 1)}, \quad (5)$$

where $E(t)$ is the evaluation value at time t, B_s and B_e are the starting and ending frames of a bounding box, respectively.

The point-to-point similarity measure between two single video frames is defined in Definition 4. The box-to-box similarity measure between two bounding boxes is defined in Definition 5. The *penalty function* determines the difference between two bounding boxes. This function value is dependent on how dissimilar these two boxes are and also what kind of query constraints they select. The value of *similarity threshold*, as defined in Definition 6, is application dependent and can be specified by users. This similarity threshold value is the error tolerance for an approximate matching and can heavily affect the performance of searching. If the similarity threshold increases, the number of qualified subsequences would increase. If the similarity threshold is equal to zero, this process becomes an exact match.

DEFINITION 4. *The point-to-point similarity measure (point-to-point distance) of a specific view between two video frames at time t_i and t_j is defined as*

$$Sp(V_1, V_2, A, t_i, t_j) = \left| E(V_1, A, t_i, t_i) - E(V_2, A, t_j, t_j) \right|, \quad (6)$$

where t_i determines the frame in sample video sequence V_1, and t_j determines the frame in query video sequence V_2.

DEFINITION 5. *The box-to-box similarity measure (box-to-box distance) of a specific view between two bounding boxes b_i and b_j is defined as*

$$S_b(V_1, V_2, A, C, b_i, b_j) = \sum_{k=1}^{n} Penalty\ (C_k, b_i, b_j), \quad (7)$$

where b_i and b_j are bounding boxes of sample video sequence V_1 and query video sequence V_2, respectively. C is the set of n query constraints for this query. Penalty(.) is the penalty function for each specified type of constraint C_k.

DEFINITION 6. *The evaluation function $E(V_1, A, t_i, t_j)$ and $E(V_2, A, t_i, t_j)$ or bounding box b_i and b_j have the similarity relation ~, if and only if*

$$S_p(V_1, V_2, A, t_i, t_j) < \tau p, \text{ for point-to-point matching or} \quad (8)$$

$$S_b(V_1, V_2, A, C, t_i, t_j) < \tau b, \text{ for box-to-box matching,} \quad (9)$$

where τp and τb are the similarity threshold of point-to-point matching and box-to-box matching, respectively.

4.3. Matching Strategies and Box-to-Box Approximate Matching

Because existence matching is easy to handle, and also partial ordering matching can be considered as the longest common subsequence problem [21], their matching algorithms will not be discussed in this paper. Therefore, we focus our matching problems only on exact sequence matching and point-to-point matching.

In the matching processes, we first divide the query sequence into its bounding box representation form. Then, these bounding boxes compare with the sample sequences in the video information system with the help of an index structure. No matter what kind of query type it is, either box-to-box or point-to-point basis, we always start our matching process from an approximate box searching approach, as shown in Algorithm 1.

In our approximate box searching approach, we use the first box with pulse or edge connection types (type 1, 2 or 3) in the query sequence as the alignment box. After a searching in index structure by the prominent point of alignment box, several candidates with the same or

ALGORITHM 1. Approximate Box Searching

Input. A sequence of bounding boxes with related box information corresponding to query sequence, an index structure, and similarity threshold.

Output. A list of similar subsequences with good-match criterion.

Method.

find first bounding box with connection type 1, 2, or 3;
search index structure by prominent point value of first bounding box to find the link list of the starting position of candidate boxes;
for *each of the candidate boxes*
if *a consecutive sequence of boxes related to the candidate starting box satisfy the query constraints* then
print out the sequence ID, starting position and similarity factor;
end-of-if
end-of-for

End-of-Algorithm Approximate Box Searching.

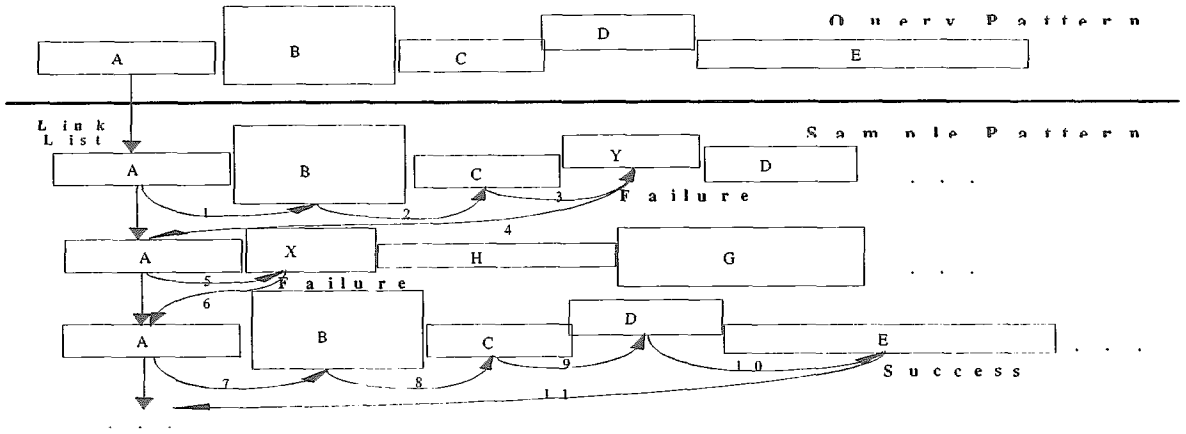


Figure 8. A box-to-box exact sequence matching example.

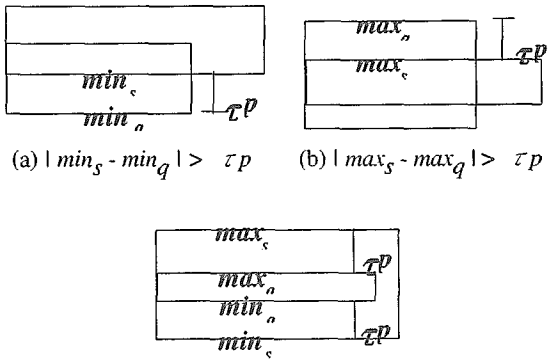
approximate prominent point value will be found. According to the specified query constraints, we can discard or prune some of the cases which do not satisfy the box-to-box similarity threshold when searching for the candidates in the index structure. An example of query constraint checking procedure for box-to-box exact sequence matching is depicted in Figure 8.

This approximate box checking algorithm acts as a filter to quickly reduce the number of possible candidates and generates a candidate set as the result of query constraint checking. Notice that false alarms are possible in these steps, but no false dismissal will occur. Further processing of the candidate set is necessary to avoid mismatching.

4.4. Point-to-Point Matching Algorithm

For a point-to-point matching, two necessary query constraints should be checked. They are the point-to-point similarity measure and the box-offset checking. After the box alignment step and the index structure searching, the next step is the query constraint checking. If we check the similarity relation for the corresponding evaluation values in the candidate box point-by-point, it will be very time-consuming. Therefore, we provide a min./max. bound similarity relation checking mechanism.

As stated in Theorem 1, we stop the searching when the difference of minimum and/or maximum values of bounding boxes of two corresponding sequences exceeds point-to-point similarity threshold τp, as shown in Figure 9. We can declare these two corresponding sequences to be dissimilar and prune this sequence from the candidate set. If all of the segmented subsequences satisfy the similar relation, we are sure that the whole sequence satisfies the similar relation.



(a) $| min_S - min_Q | > \tau p$ (b) $| max_S - max_Q | > \tau p$

(c) both of them

Figure 9. Similarity relation checking of bounding boxes.

THEOREM 1. *There exists at least one evaluation value in the query box that can not satisfy the similarity relation, if*

$$| min_S - min_Q | > \tau p \text{ or,} \quad (10)$$

$$| max_S - max_Q | > \tau p, \quad (11)$$

where min_S and max_S are the minimum value and maximum value of a sample box, respectively; min_Q and max_Q are the minimum value and maximum value of a query box, respectively. τp is the point-to-point similarity threshold.

5. CONCLUSIONS

By providing multi-level content-based retrieval, applications of digital video are broad in many aspects. Video records the changes of scenes according to time. Related change of objects between different frames provides much information about the behavior of these objects in the video. These time-series changes of video objects are useful for dynamic scene and motion analysis.

In this paper, we have presented the design of an approximate video content matching algorithm. The idea is to extract video contents via low level feature extraction and/or high level semantic retrieval mechanisms according to a specific point of view, then segment video contents into bounding boxes via a box segmentation mechanism by their time series feature values. With the help of indexing mechanism using prominent index points, the searching speed is faster than the sequential scanning method. A video information prototype system example and several experimental results show how these mechanisms work. Notice that time-series data indexing and matching mechanism can also be applied to many other applications, such as banking, policy decisions, inventory control, and

scientific databases, where the history and prediction are important.

REFERENCES

[1] A. Hampapur, R. Jain and T. Weymouth, Digital Video Segmentation, in *Proceedings, ACM Multimedia, San Francisco, USA, 1994*, pp. 357-364.

[2] H. J. Zhang, S. Y. Tan, S. W. Smoliar and G. Yihong, Automatic Parsing and Indexing of News Video, *Multimedia Systems*, **2**, 1995, 256-266.

[3] E. Oomoto and K. Tanaka, OVID: Design and Implementation of a Video-Object Database System, *IEEE Trans. on Knowledge and Data Engineering*, **5**, No. 4, Aug. 1993, 629-643.

[4] S. W. Smoliar and H. Zhang, Content-based Video Indexing and Retrieval, *IEEE Multimedia*, Summer 1994, 62-72.

[5] C. W. Chang and S. Y. Lee, Indexing and Approximate Matching for Content-based Time-series Data in Video Database, in *Proceedings, First Intl. Conference on Visual Information Systems, Melbourne, Australia, 1996*, pp. 567-576.

[6] S. Y. Lee and H. M. Kao, Video Indexing - An Approach based on Moving Object and Track, in *Proceedings, SPIE Storage and Retrieval for Image and Video Databases, Vol. 1908, 1993*, pp. 25-36.

[7] Y. F. Day, S. Dagtas, M. I. A. Khokhar and A. Ghafoor, Object-Oriented Conceptual Modeling of Video Data, in *Proceedings, IEEE 11th Intl. Conference on Data Engineering, Taipei Taiwan,1995*, pp. 401-408.

[8] A. Nagasaka and Y. Tanaka, Automatic Video Indexing and Full-Video Search for Object Appearances, in *Visual Database Systems II* (E. Knuth and L. M. Wegner ed.), pp. 113-127, 1992.

[9] A. D. Narasimhalu, Special Section on Content-based Retrieval, *Multimedia Systems*, **3**, Feb. 1995.

[10] A. Yoshitaka, S. Kishida, M. Hirakawa and T. Ichikawa, Knowledge-Assisted Content-Based Retrieval for Multimedia Databases, in *Proceedings, IEEE Intl. Conference on Multimedia Computing and Systems, Boston, 1994*, pp. 131-139.

[11] T. D. C. Little et al., A Digital On-Demand Video Service Supporting Content-Based Queries, in *Proceedings, ACM First Intl. Conference on Multimedia, Anaheim, California*, 1993, pp. 427-436.

[12] R. Bach, S. Paul and R. Jain, A Visual Information Management System for the Interactive Retrieval of Faces, *IEEE Trans. on Knowledge and Data Engineering*, **4**, No. 4, Aug. 1993, 619-628.

[13] M. Flickner et al. Query by Image and Video Content: The QBIC System, *IEEE Computer*, Sep. 1992, 23-32.

[14] F. Arman, R. Depommier, A. Hsu and M. Y. Chiu, Content-based Browsing of Video Sequences, in *Proceedings, ACM Intl. Conference on Multimedia, San Francisco, USA, 1994*, pp. 97-103.

[15] Y. Tonomura, A. Akutsu, Y. Taniguchi and G. Suzuki, Structured Video Coomoputing, *IEEE Multimedia*, Fall 1994, 34-43.

[16] C. Faloutsos, M. Ranganathan and Y. Manolopoulos, Fast Subsequence Matching in Time-Series Databases, in *Proceedings, ACM SIGMOD, Minneapolis, USA, 1994*, pp. 419-429.

[17] C. W. Chang, K. F. Lin and S. Y. Lee, The Characteristics of Digital Video and Considerations of Designing Video Databases, in *Proceedings, ACM Fourth Intl. Conference on Information and Knowledge Management CIKM'95, Baltimore, Maryland, 1995*, pp. 370-377.

[18] K. Wakimoto, M. Shima, S. Tanaka and A. Maeda, Content-based Retrieval Applied to Drawing Image Database, in *Proceedings, SPIE, Vol. 1908, 1993*, pp. 74-84.

[19] D. Comer, The Ubiquitous B-Tree, *ACM Computing Surveys*, **11**, No. 2, Jun. 1979, 121-137.

[20] T. Ito and M. Kizawa, Hierarchical File Organization and Its Application to Similar String Matching, *ACM Trans. on Database Systems*, **8**, No. 3, Sep. 1983, 410-433.

[21] Y. P. Wang and T. Pavlidis, Optimal Correspondence of String Subsequences, *IEEE Trans. on Pattern Analysis Machine Intelligent*, **12**, No. 11, Nov. 1990, 1080-1087.

[22] A. Yoshitaka and T. Ichikawa, A Survey on Content-based Retrieval for Multimedia Databases, *IEEE Trans. on Knowledge and Data Engineering*, **11**, No. 1, 1999, 81-93.

[23] B. Furht, S. W. Smoliar and H. Zhang, *Video and Image Processing in Multimedia Systems*, Kluwer Academic Publishers, 1995.

[24] Y. Gong, *Intelligent Image Databases: Toward Advanced Image Retrieval*, Kluwer Academic Publishers, 1998.

[25] A. K. Elmagarmid, A. A. Helal, A. Joshi and M. Ahmed, *Video Database Systems: Issue, Products and Applications*, Kluwer Academic Publishers, 1997.

[26] Y. Alp Aslandogan and C. T. Yu, Techniques and Systems for Image and Video Retrieval, *IEEE Trans. on Knowledge and Data Engineering*, **11**, No. 1, 1999, 56-63.

[27] S. Shekhar, S. Chawla and S. Ravada, Spatial Databases-Accomplishments and Research Needs, *IEEE Trans. on Knowledge and Data Engineering*, **11**, No. 1, 1999, 45-55.

[28]C. S. Jensen and R. T. Snodgrass, Temporal Data Management, *IEEE Trans. on Knowledge and Data Engineering*, **11**, No. 1, 1999, 36-44.

[30] W. Al-Khatib, A. Ghafoor and P. B. Berra, Semantic Modeling and Knowledge Representation in Multimedia Databases, *IEEE Trans. on Knowledge and Data Engineering*, **11**, No. 1, 1999, 64-80.

5 結論

本篇論文提出一個結合離線和線上功能的整合系統，離線子系統利用多維空間索引，根據關鍵畫面先找出與查詢影像相似的場景，而線上子系統即時比對場景的每個視訊畫面，針對使用者輸入的查詢影像找出相似視訊畫面的部份內容。此系統具有下列五大特性：(1) *可適性(adaptability)*，系統可視不同的網路頻寬限制與應用環境，彈性選擇最適當模式以增加整體效益；(2)*擴充性(scalability)*，主從式的開放架構的使其易於建構在網路中；(3)*有效性(efficiency)*，無論離線或線上處理皆可在壓縮領域下運算且完全自動化；(4) *可行性(effectiveness)*，線上多解析度的擷取方式符合視訊壓縮標準的規格，例如 MPEG 提供 SNR、spatial 及 temporal scalabilit 的功能；(5)*高彈性(flexibility)*，系統提供區塊的近似與部份比對法則，同時兼顧查詢的效能和品質。本系統提供網路上搜尋視訊資料的設計架構，但目前尚有些限制，例如：區塊比對方法無法查詢物件的放大縮小與旋轉等情形，且無法分離出前景與背景物件。
除了適用於網路環境上的視訊資料內容查詢系統，本系統同樣地可應用於文字、影像、聲音等多媒體資料庫查詢系統。未來的研究方向考慮在網路查詢系統中再加入相關性回饋(relevant feedback)的功能，以自動學習使用者在視覺感官上相似的規則，更有效地找出使用者真正想要的視訊資料。

國科會計劃編號 NSC89-2213-E-009-015

參考文獻

[1] N. Beckmann, H. -P. Kriegel, R. Schneider and B. Seeger, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," in *Proc. of ACM SIGMOD*, Atlantic City, USA, pp.322-331, May 1990.

[2] M. L. Cascia and E. Ardizzone, "JACOB: just a content-based query system for video databases," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 7-10, 1996.

[3] S. F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC -DCT compressed video, *IEEE Journal on Selected Areas in Communications: Special Issue on Intelligent Signal Processing*, vol. 13, no. 1, pp.1-11, Jan. 1995.

[4] P. J. Cheng and W. P. Yang, "A new content-based video retrieval system - data model and query processing," in *Proc. of W rkshop on Softwar Engineering and database Systems, International Computer Symposium*, pp. 1 -122, 1998.

[5] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," in *Proc. of ACM SIGMOD*, Minnesota, USA, 1994.

[6] F. Idris and S. Panchanathan, "Review of image and video indexing technique," *Journal of Visual Communication and Image Representation*, vol. 8, no. 2, pp.146-166, 1997.

[7] V. Kobla and D. Doermann, "Indexing and retrieval of the MPEG compressed video," *Journal of Electronic Imaging*, vol. 7, no. 2, pp.294-307, April 1998.

[8] S. Lee, M. Yang and J. Chen, "Signature file as a spatial filter for iconic image databases," *Journal of Visual Languages and Computing*, vol. 3, pp. 373-397, 1992.

[9] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker and C. Faloutsos, "The QBIC project: Querying images by content using color, texture, and shape," in *Storage and Retrieval for Image and Video Databases*, SPIE, vol. 1908, pp.173-187, February 1993.

[10] A. P. Pentland, R. W. Picard and S. Sclaroff "Photobook: Content -based manipulation of image databases," *International Journal of Computer Vision*, vol. 18, no. 3, pp.233-254, 1996.

[11] J. R. Smith and S. F. Chang, "VisualSEEK: a fully automated content-based image query system," *ACM Multimedia*, Boston, MA, November 1996.

[12] J. R. Smith, "VideoZoom Spati -temporal Video Browser," *IEEE Transactions on Multimedia*, vol. 1, no. 2, pp. 157-171, June 1999.

[13] B. L. Yeo and B. Liu, "On the extraction of DC sequence from MPEG compressed video," *IEEE International Conference on Image Processing*, vol. 2, pp. 260-263, 1995.

[14] A. Yoshitaka and T. Ichikawa, "A survey on content-based retrieval for multimedia databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 1, pp. 81-93, 1999.

[15] H. J. Zhang, C. Y. Low, Y. Gong, S. W. Smoliar and S. Y. Tan, "Video parsing compressed data," in *Proc. Of SPIE: Image Video Processing* II 2182, pp.142-149, 1994.