Efficient Video Indexing Scheme for Content-Based Retrieval

Hyun Sung Chang, *Member, IEEE*, Sanghoon Sull, *Member, IEEE*, and Sang Uk Lee, *Senior Member, IEEE*

***Abstract*— Extracting a small number of key frames that can abstract the content of video is very important for efficient browsing and retrieval in video databases. In this paper, the key frame extraction problem is considered from a set-theoretic point of view, and systematic algorithms are derived to find a compact set of key frames that can represent a video segment for a given degree of fidelity. The proposed extraction algorithms can be hierarchically applied to obtain a tree-structured key frame hierarchy that is a multilevel abstract of the video. The key frame hierarchy enables an efficient content-based retrieval by using the depth-first search scheme with pruning. Intensive experiments on a variety of video sequences are presented to demonstrate the improved performance of the proposed algorithms over the existing approaches.**

***Index Terms*— Content-based retrieval, key frame extraction, key frame hierarchy, video database, video indexing.**

I. Introduction

DUE to their effectiveness for conveying audio-visual information, digital video data have become very popular through the Internet and digital libraries. At the same time, the rapid increase in digital video data invokes the need for efficient retrieval and browsing. Recently, several approaches, based on key frames, have been proposed for video retrieval and browsing [2]–[9]. Key frames are a small set of images that can represent the visual content of a video. They can be used to compute the similarity between two video sequences, as well as to browse the video based on its content.

The first step commonly taken for content-based video processing is to divide a given video into shots. A shot is considered as a set of successive similar frames. This situation happens, for example, when the video is filmed either from a fixed camera position or using coherent camera motion, such as panning, rotation, and zooming [10]. A variety of methods [11]–[13] were proposed to detect the shot boundaries. After the shot boundaries are identified, most of the existing works for video abstraction generally go through the following two steps: first, select the key frames in each shot [2]–[6], and then cluster the similar shots based on the key frames [7]–[9] to construct the hierarchical or transition representation of video. In this paper, we refer to the former as the *key frame extraction* (KFE) *at the shot level* and the latter as *video representation*. The hierarchical or transition structures can provide a user-friendly summarized view of the video content to the users. In the existing work, most of the attention is paid to the development of browsing methods, while there have been few studies for retrieval.

In this paper, we propose a novel approach for obtaining a compact representation of video, also useful for the retrieval. The KFE problem is modeled as choosing a compact set of samples (key frames) among many data points (frames in a video shot), while keeping the distortion[1] less than a given threshold, which is analogous to the vector quantization scheme [14]. Then, systematic extraction algorithms based on the point set theory are presented. In the proposed algorithms, the combinatorial selection of a (sub)minimal set of key frames under the given fidelity constraint yields (sub)optimal results in terms of rate[2]-distortion (R-D) performance. By using the combinatorial property, the proposed extraction methods can be hierarchically applied to higher levels, starting from the frames in each shot, yielding a tree-structured key frame hierarchy. The key frame hierarchy is a multilevel abstract of a video, in which each level represents the whole video content at different level of details. It also enables an efficient frame retrieval, using the depth-first search (DFS) with pruning.

This paper is organized as follows. In the next section, we propose a new measure for the fidelity of a given key frame set for representing a video segment. We also describe the condition that should be satisfied by the key frame set. In Section III, we present graph-based algorithms that can search a set of key frames with (sub)optimal R-D performance. In Section IV, we describe how the proposed methods can be used for extracting the key frames at the shot level and constructing a structured index of video. In Section V, we demonstrate the performance of the proposed methods through the experiments. Section VI concludes this paper.

II. Measure of the Goodness of Selected Key Frames

Let us consider a key frame set extracted for a video segment[3] by an arbitrary KFE method. For effective video browsing and retrieval, the selected key frames should be able to represent the content of the video segment. Although

Manuscript received October 31, 1998; revised August 1, 1999. This paper was recommended by Guest Editor S.-F. Chang.

H. S. Chang is with the Radio & Broadcasting Technology Laboratory, Electronics and Telecommunications Research Institute, Taejon 305-350 Korea.

S. Sull is with the School of Electrical Engineering, Korea University, Seoul 136-701 Korea.

S. U. Lee is with the School of Electrical Engineering, Seoul National University, Seoul 151-742 Korea.

Publisher Item Identifier S 1051-8215(99)09586-5.

[1] Distortion comes from excluding some frames from the set of key frames.

[2] To measure the compactness of a set of key frames, we define rate as the ratio of the number of key frames.

[3] In the literature, key frames are mainly defined on a shot, but we do not restrict ourselves to the shot level.

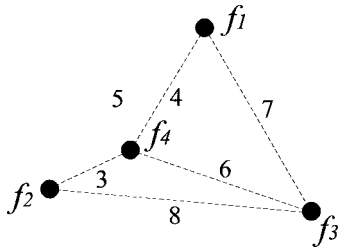


Fig. 1. An example to explain the semi-Hausdorff distance. Each number on the line connecting two points indicates the distance between them.

there have been many attempts [2]–[6] to select key frames, a proper criterion for the goodness of the key frames, which can describe quantitatively their fidelity, has not been reported yet. In this section, we propose a new measure based on semi-Hausdorff distance to evaluate the fidelity of a key frame set and present the necessary and sufficient condition for the key frame set to satisfy. We begin by briefly reviewing the definition of the semi-Hausdorff distance for the sake of completeness.

A. Semi-Hausdorff Distance

Let $(\mathcal{X}, d)$ be a metric space, where d is a predefined distance function. For two point sets $A, B \subset \mathcal{X}$, the semi-Hausdorff distance from A to B, denoted by $d_{\mathrm{SH}}(A, B)$, is defined as [15]

$$\begin{aligned} d_{\mathrm{SH}}(A,B) &= glb\{\epsilon | A \subset U(B,\epsilon)\} \\ U(B,\epsilon) &= \bigcup_{x \in B} B_d(x,\epsilon) \\ B_d(x,\epsilon) &= \{y | d(x,y) \le \epsilon\} \end{aligned} \tag{1}$$

where glb represents the greatest lower bound.

Considering the case shown in Fig. 1, let us denote $\{f_1, f_2, f_3, f_4\}$ by A and one of its subsets, $\{f_1, f_4\}$, by B. Suppose two circles C_1 and C_4 with their radii close to zero, centered at f_1 and f_4, respectively. Initially, the two points f_1 and f_4 are inside the circles, while f_2 and f_3 are not. If we expand the radii of the circles with the same speed, the point f_2 is barely included into the circles when the radii become three. Continuing to expand the radii, f_3 will be also covered by C_4, whose radius has grown to six, and there is no uncovered point. Therefore, by the definition in (1), $d_{\mathrm{SH}}(A, B) = 6$.

B. A New Measure of the Fidelity of a Set of Key Frames

Assuming that each frame corresponds to one feature point, we can apply the definition (1) to the image feature space $(\mathcal{I}, d)$ of our interest. Visualizing that all the frames in a video segment S are scattered points in $\mathcal{I}$, our goal is to optimally extract a set of key frames R from S. Here, we propose $d_{\mathrm{SH}}(S, R)$ as a measure for the fidelity of the selected key frames R, based on the following property (see Appendix A for its proof).

Property 1: $d_{\mathrm{SH}}(S, R) > \epsilon$ if and only if $\exists f \in S$ such that $\min_{\hat{f} \in R} \; d(f, \hat{f}) > \epsilon$.

Although (1) appears quite simple and trivial, Property 1 implies that $d_{\mathrm{SH}}(S, R)$ can be used as a tight measure to determine how well a given set of key frames represents the entire video segment.

Returning to the previous example in Fig. 1, assume that $S = \{f_1, f_2, f_3, f_4\}$ and $R = \{f_1, f_4\}$. Then, the fact that $d_{\mathrm{SH}}(S, R)$ is equal to six can be interpreted as follows. The two frames f_1 and f_4 in S are perfectly represented by the key frame set $R = \{f_1, f_4\}$, which is evident. The remaining two frames f_2 and f_3 should be also represented by R. f_2 can be approximated to f_4, rather than to f_1, and the approximation error is $d(f_4, f_2) = 3$. Similarly, f_3 is approximated to f_4 with an error of six, and thus the error resulting from the approximation of S by R is six, which is equal to $d_{\mathrm{SH}}(S, R)$.

Generally $d_{\mathrm{SH}}(S, R)$ is decreased by utilizing more frames as key frames. For example, if $R = S$, then $d_{\mathrm{SH}}(S, R) = 0$. Therefore, we are interested in selecting the minimal set of key frames, while maintaining $d_{\mathrm{SH}}(\cdot)$ below a predefined threshold ϵ.

The metric $d(\cdot)$ defined over $\mathcal{I}$ is assumed to have the following properties:

1) $d(f_i, f_j) \ge 0, \forall f_i, f_j \in \mathcal{I}$, where the equality holds if and only if f_i and f_j occupy the same point in $\mathcal{I}$;
2) $d(f_i, f_j) = d(f_j, f_i), \forall f_i, f_j \in \mathcal{I}$;
3) $d(f_i, f_j) \le d(f_i, f_k) + d(f_k, f_j), \forall f_i, f_j, f_k \in \mathcal{I}$ (triangle inequality).

The majority of the existing dissimilarity measures satisfy the metric properties and can be used for $d(\cdot)$. Some of the popular measures are described below.

1) Color Histogram: Color is a very effective visual attribute for the description of a scene. The color indexing scheme developed by Swain and Ballard [16] is known to work well, even if there exist some variations, such as a change in view position and partial occlusion. To evaluate the similarity between two images f_i and f_j of size $M \times N$, [16] intersects two color histograms H_i and H_j, which capture the global distribution of colors in f_i and f_j, respectively, in the following way:

$$s(f_i, f_j) = \sum_{k=1}^{n} \min \{H_i(k), H_j(k)\} \tag{2}$$

where n denotes the number of bins in each histogram. To express the dissimilarity between the two images

$$d(f_i, f_j) = 1 - \frac{1}{M \times N} \sum_{k=1}^{n} \min \{H_i(k), H_j(k)\} \tag{3}$$

within the range of [0,1], can be used, instead of (2). Note that (3) also satisfies the metric properties.

The indexing schemes using color have been extended and refined by many researchers [17]–[20].

2) Correlation: Among the most frequently used measures to evaluate the correlation between two images are the L_p norms, given by

$$d_p(f_i, f_j) = \left(\sum_{m=1}^{M} \sum_{n=1}^{N} |f_i(m,n) - f_j(m,n)|^p \right)^{1/p}, \quad p = 1, 2, \ldots, \infty \tag{4}$$

for two images f_i and f_j of size $M \times N$. The cases of $p = 1$ and $p = 2$ are particularly of interest; They are often called "city block distance" and "Euclidean distance," respectively. They are known to provide good indications of the dissimilarity, in spite of the simplicity. All of the norms satisfy the metric property.

It is often the case that the distance functions employing the luminance projection vectors, instead of full images, are used to reduce the dimension of feature space [3]. The luminance projection vectors for the nth row and the mth column, denoted by l_n^r and l_m^c, respectively, are

$$l_n^r(f) = \sum_{m=1}^{M} Lum\{f(m,n)\} \tag{5}$$

and

$$l_m^c(f) = \sum_{n=1}^{N} Lum\{f(m,n)\}. \tag{6}$$

Then, the distance function $\hat{d}_p(\cdot)$, normalized to [0,1], is defined as

$$\hat{d}_p(f_i, f_j) = \frac{1}{K}\left(\sum_{n=1}^{N}\left|\frac{1}{M}(l_n^r(f_i) - l_n^r(f_j))\right|^p + \sum_{m=1}^{M}\left|\frac{1}{N}(l_m^c(f_i) - l_m^c(f_j))\right|^p\right)^{1/p} \tag{7}$$

where K is a normalizing constant. The performance of the metric (7) is reported to be comparable to that of using (4).

Although the dissimilarity measures described above are good candidates for $d(\cdot)$, other metric measures can be found. The majority of the nonmetric measures can be slightly modified to have the metric properties with similar performance.

C. Necessary and Sufficient Condition for a Set of Key Frames

According to Property 1, in order to approximate the entire video segment S by a key frame set R within an error bound ϵ, the following condition should be met:

$$d_{\mathrm{SH}}(S, R) \le \epsilon \tag{8}$$

which is equivalent to

$$\bigcup_{f_i \in R} C_i = S \tag{9}$$

where

$$C_i \triangleq \{f_j \in S | d(f_i, f_j) \le \epsilon\}. \tag{10}$$

If we set ϵ, the supremum of $d_{\mathrm{SH}}(\cdot)$, to four in Fig. 1, only five out of 16 possible cases, which are listed in Table I, are allowed because other cases do not satisfy (8). Among them, only the first one is optimal, in the sense that the cardinality of R, denoted by $\mathrm{card(R)}$, is the smallest, while satisfying the condition (8).

From Property 1, it is obvious that the condition (9) also serves as a sufficient condition, not just a necessary condition, for a subset of S to be a set of key frames. For example, in Fig. 2, in which each circle represents C_3, C_6, C_9, and C_{12}, respectively, a subset $\{f_3, f_6, f_9, f_{12}\}$ is qualified to be a set of key frames, since the union of them covers S.

TABLE I
KEY FRAME SELECTION SCHEMES ALLOWED FOR $d_{\mathrm{SH}}(\cdot)$ NOT LARGER THAN FOUR IN Fig. 1. THE SUPERSCRIPT c INDICATES THE COMPLEMENTARY OPERATION

R	R^c	$d_{SH}(S,R)$	Critical Pair
$\{f_3, f_4\}$	$\{f_1, f_2\}$	4	(f_4, f_1)
$\{f_1, f_2, f_3\}$	$\{f_4\}$	3	(f_2, f_4)
$\{f_1, f_3, f_4\}$	$\{f_2\}$	3	(f_4, f_2)
$\{f_2, f_3, f_4\}$	$\{f_1\}$	4	(f_4, f_1)
$\{f_1, f_2, f_3, f_4\}$	$\emptyset$	0	–

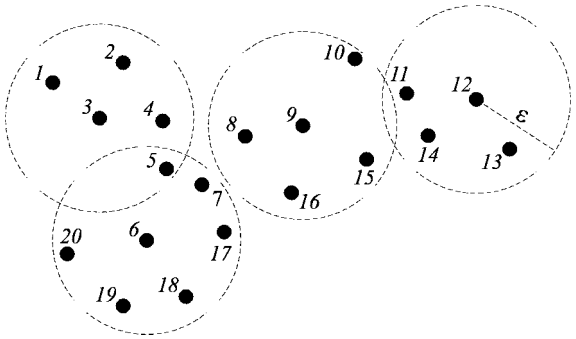


Fig. 2. Twenty frames in feature space $(\mathcal{I}, d)$ corresponding to a video segment S. Each number denotes the frame number.

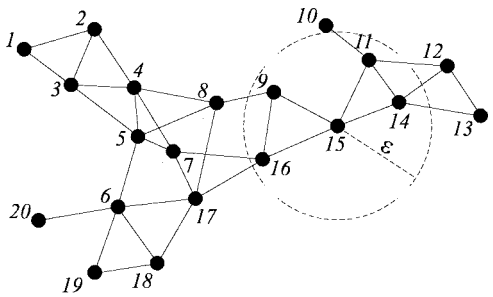


Fig. 3. Proximity graph generated by the parameter ϵ for Fig. 2.

III. KEY FRAME EXTRACTION ALGORITHMS

In this section, we present an algorithm that can search an optimal key frame set in terms of R-D performance. In other words, it chooses the minimal set of key frames, while satisfying the condition (8) simultaneously. Then, to reduce the computational cost, based on the graph theory, we also describe suboptimal greedy algorithms whose complexities are much alleviated.

A. Algorithm Based on Graph Theory

First, we construct a proximity graph as shown in Fig. 3, in which each frame in a video segment S corresponds to a vertex and two vertices, whose distance or cost is less than ϵ, are connected by an edge. Then, the C_i defined in (10) becomes the set containing f_i itself and the neighboring vertices. Examples are illustrated in Table II. Letting $\mathcal{C}_0$ be $\{C_i | f_i \in S\}$, $\mathcal{C} \subset \mathcal{C}_0$ satisfying $\cup\, \mathcal{C} = S$ is referred to as a

TABLE II
$\{C_i\}$ FOR EACH $f_i \in S$ IN FIG. 3

C_i	Elements
C_1	f_1, f_2, f_3
C_2	f_1, f_2, f_3, f_4
C_3	f_1, f_2, f_3, f_4, f_5
⋮	⋮
C_{20}	f_6, f_{19}, f_{20}

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}
C_1	x	x	x																	
C_2	x	x	x	x																
C_3	x	x	x	x	x															
C_4		x	x	x	x		x	x												
C_5			x	x	x	x	x	x									x			
C_6					x	x	x										x	x	x	x
⋮																				
C_{19}						x												x	x	x
C_{20}						x													x	x

Fig. 4. Cover table constructed for Fig. 2.

cover of S. Finally, our goal is to find the minimal cover of S in the constructed proximity graph.

B. An Optimal Approach

Once the cover table is constructed as shown in Fig. 4, the optimal solution for the minimum covering problem can be found using parts of the Quine–McCluskey algorithm,[4] which is quite popular and well known in logic design [21]. The cover table, in which the rows and the columns represent all $\{C_i\}$'s and all $\{f_j\}$'s to be covered, respectively, contains a "$\times$" at the intersection of row C_i and column f_j if $f_j \in C_i$. Then, the Quine–McCluskey algorithm utilizes the following relations among the rows and the columns to reduce the table size.

1) If two rows C_i and C_j are related by $C_i \subset C_j$, then it is said that row C_i is dominated by row C_j, and row C_i can be deleted.
2) If every row that takes column f_j as one of its elements also contains the column f_i, then column f_i is said to dominate column f_j and can be deleted.

For example, the rows C_1 and C_2 in Fig. 4 can be deleted because they are dominated by row C_3. In our case, the use of dominance relations greatly reduces the number of columns to be covered. After all possible simplifications have been made, the algorithm adopts the branch-and-bound search to select the minimal number of rows whose union covers all of the columns.

C. A Greedy Approach

To find a minimal cover by using the Quine–McCluskey method is NP-complete. Therefore, the proposed optimal method may be impractical for a video segment containing large number of frames. To solve this problem, we employ a greedy method whose computational cost is significantly lower with slight degradation in the performance.

[4] Also known as "tabular method" in logic design literature.

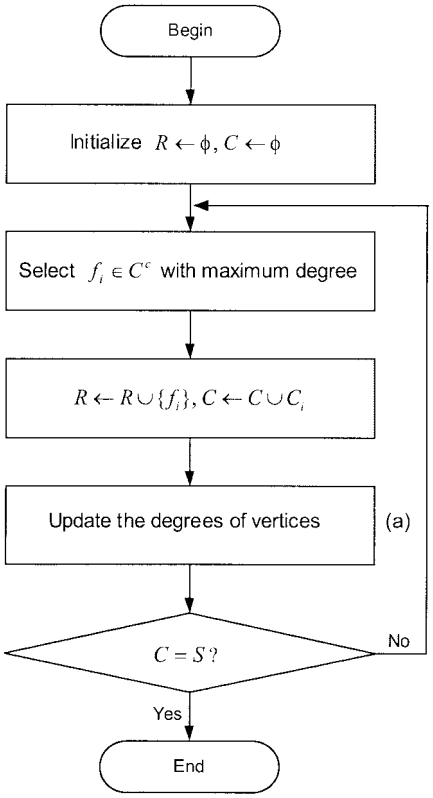


Fig. 5. Algorithm I: greedy algorithm for the fidelity-constrained KFE problem. The superscript "c" indicates the complementary operation.

Let R and C be a set of key frames and the set of frames covered by R, respectively. Then, set R finally obtained from Algorithm I in Fig. 5 satisfies (9), qualifying to be a key frame set. In Algorithm I, the degree of a vertex f_i, denoted by $deg(f_i)$, is the number of frames to be newly included into C, in case of selecting f_i as a key frame. Since C is empty at the start, it is initially set to $card(C_i)$.

It is often the case that the complete mapping between each frame in S and the representative key frame in R is required. For instance, the mapping keeps the temporal orders among the key frames, which is desirable for browsing. In Fig. 2, the effective browsing is achieved by arranging the key frames in the order of f_3–f_6–f_9–f_{12}–f_9–f_6, although there are only four key frames obtained. It is a representation of the original frames in their sequential order by using the key frames. During the mapping, the nearest neighbor (NN) rule is applied to arbitrate possible ambiguities such as f_5 in Fig. 2, which is covered by both C_3 and C_6. In other words, a key frame f_i has its coverage $\hat{C}_i = \{f_j \in S | d(f_i, f_j) \leq d(f_k, f_j),\ \forall f_k \in R\}$ with an associated value $\delta_i = \max_{f_j \in \hat{C}_i} d(f_i, f_j)$ (note that $\delta_i \leq \epsilon$). The NN rule eventually completes the definition of the extraction operator $\mathcal{E}$ as a mapping of all the frames in S into the key frames in R, which is expressed as

$$\mathcal{E}: S \longrightarrow R \tag{11}$$

or simply

$$R = \mathcal{E}(S). \tag{12}$$

```
FOR each f_j ∈ C_i,
    IF f_j has not been covered yet THEN
        f_j is covered by f_i.
        FOR each f_k ∈ C_j,
            decrement deg(f_k) by 1.
        ENDFOR;
    ELSE /* f_j has been covered by a frame f_m */
        IF d(f_i, f_j) ≤ d(f_m, f_j) THEN
            f_j is covered by f_i.
        ENDIF;
    ENDIF;
ENDFOR;
```

Fig. 6. Algorithm to implement "Update of degree + Nearest neighbor rule" under the assumption that f_i is selected as a key frame.

The NN rule, combined with the degree updating process, is implemented by the simple algorithm in Fig. 6, replacing part (a) in Algorithm I. It can be easily shown that the overall procedures of Algorithm I require $O(V^2)$ in time and space, where V denotes the number of vertices (frames) in the proximity graph.

D. Rate-Constrained KFE Problem

There are some cases in which the control on the number of key frames (i.e., rate) is more important than that on the fidelity, particularly for the system with limited resources. Although Algorithm I could adjust the number of key frames by an appropriate fidelity constraint, it is not easy to translate the constraint on rate into that on fidelity.

Let the desired number of key frames be K. For $K = 1$, it is quite simple to find the key frame set minimizing the distortion measure (i.e., the semi-Hausdorff distance). However, the cases of $K \geq 2$ are not straightforward. To address this problem, we resort to greedy strategy again, as depicted in Algorithm II of Fig. 7.

Algorithm II can also solve the original problem given a fidelity constraint; just by stopping with the smallest K the fidelity constraint is met. As will be shown in Section V, Algorithm II yields an R-D performance comparable to that of Algorithm I, while requiring a little more time.

IV. Video Indexing Scheme

In this section, we first discuss the methods of extracting the key frames at the shot level. Then, we propose a tree-structured key frame hierarchy obtained from the hierarchical application of the KFE algorithm as an efficient video index. In this section, it is assumed that the shot boundaries were identified in advance and the series of shots $\{S_i\}_{i=1}^{I}$ is given to us.

A. Key Frame Extraction at the Shot Level

A video shot typically consists of more than hundreds of frames, making the content-based search difficult task. However, since the frames are highly correlated (similar) in temporal domain, the KFE schemes can select a small set of representative and decorrelated frame samples, reducing the search space significantly.

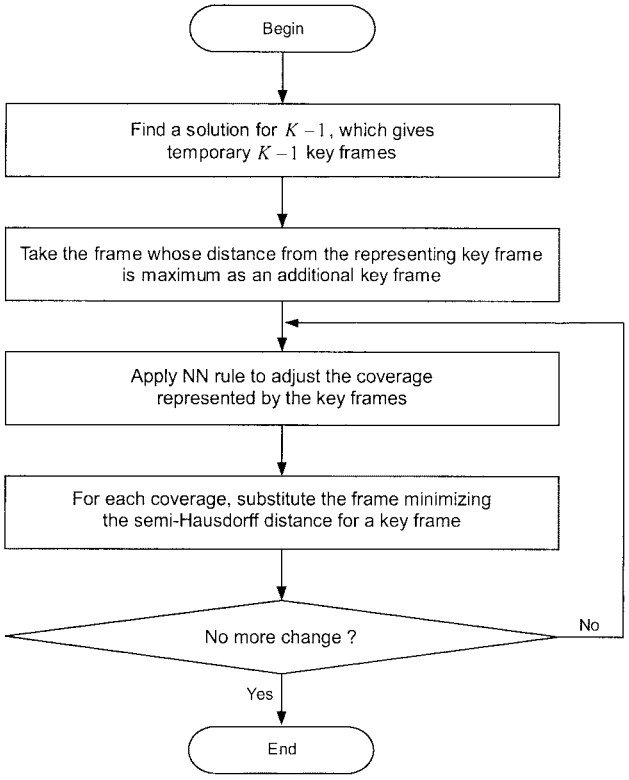


Fig. 7. Algorithm II: algorithm for the rate-constrained KFE problem. The desired number of key frames is K $(K \geq 2)$.

The simplest approach to the KFE problem is to select an arbitrary frame in each shot [2]. Later, Kobla *et al.* [22] consider the minimum bounding rectangle (MBR) of point trajectory in the image feature space (similar to $\mathcal{I}$ in this paper) and present a simple scheme that takes the point closest to the center of MBR(S) as a representative of the shot S. However, the schemes, which allow only one key frame for each shot, often lose too much visual information to represent the entire shot faithfully.

More advanced methods to take into account of the temporal variations within individual shots have been also proposed. Yeung and Liu [3] search the key frames in sequential manner. In [3], the first frame in a shot is always selected as one of the key frames, and through serial search the next frame, not covered by the last selected key frame, is taken as another. Assuming that each key frame represents a contiguous interval in a shot, Lagendijk *et al.* [4] proposed a key frame allocation method. They attempt to optimally determine the boundaries of the intervals and the location of the key frame within each interval, similar to the Lloyd–Max algorithm in the design of a scalar quantizer [14]. The approach in [4] appears reasonable for browsing. However, since it is constrained by the sequential order of frames,[5] it may not yield good

[5]In the formulation of the object function, they implicitly assume $d(f_i, f_j) = d(f_i, f_k) + d(f_k, f_j)$, $i \leq k \leq j$, where i, j, k denotes each time index. But, in general cases, $d(f_i, f_j) \leq d(f_i, f_k) + d(f_k, f_j)$.

performance, in terms of R-D. DeMenthon *et al.* [23] consider the KFE problem as fitting the curvature of point trajectory in the feature space, in which the curvature characteristic frames are iteratively selected as key frames. Similarly to [4], the key frame set found by [23] is believed to be quite good for the perceptual browsing, but it is not good in terms of R-D performance. A KFE algorithm based on unsupervised clustering was proposed by Zhuang *et al.* [5]. The algorithm is similar to the proposed ones in the sense that a key frame is allowed to represent several disjoint intervals in a shot. However, in [5], the center of each cluster, which is not a frame point in general, is not guaranteed to cover all the frames in the same cluster within the bound of their extraction parameter δ, which is different from ours. Deng and Manjunath [24] attempt to take into account of the motion information simultaneously, in which every intracoded frame (I-frame) is employed for color and texture information, and all the intercoded frames (P- and B-frame) for motion information.

The extraction algorithms proposed in Section III are applied to this problem, by considering each shot S_i. With the fidelity constraint ϵ_1, it is formulated as

$$R = \bigcup_{i=1}^{I} \mathcal{E}(S_i; \epsilon_1). \tag{13}$$

To find the (sub)minimal set of key frames for the fidelity, the proposed algorithm uses the combinatorial selection scheme. But, by the mapping process, the sequential order of the frames is also kept, which is desirable for the browsing.

B. Structured Index of Video

To find similar frames in a video for a given query image, all the key frames could be searched sequentially. But the size of the key frame set for a long video sequence might be still too large, requiring more efficient indexing scheme.

The proposed KFE algorithms are directly extensible to higher levels in the following ways:

$$R^k = \mathcal{E}(R^{k-1}; \epsilon_k), \qquad k = 2, 3, \cdots, L \tag{14}$$

where R^k is the set of the kth level key frames and L denotes the total number of levels in the hierarchy. Let us denote the frame set of the original video by R^0.[6] First, the operation in (13) extracts R^1 (simply denoted by R in previous sections), suppressing $d_{\mathrm{SH}}(R^0, R^1)$ not larger than ϵ_1. Next, R^2 is obtained from R^1, under the fidelity constraint $d_{\mathrm{SH}}(R^1, R^2) \leq \epsilon_2$, and so forth. These bottom-up procedures finally build the tree-structured key frame hierarchy for the video, as shown in Fig. 8.

For the sake of convenience, let us define some notations in the key frame hierarchy as:

f_i^k an arbitrary (or the ith) frame in the set R^k;
T_i^k subtree rooted at f_i^k;
$\hat{C}_i^k$ direct coverage of f_i^k, that is, $T_i^k \cap R^{k-1}$;

[6] $R^0 = \cup_{i=1}^{I} S_i$ is assumed to exclude the dissolves, or gradually scene changing parts, from the original video.

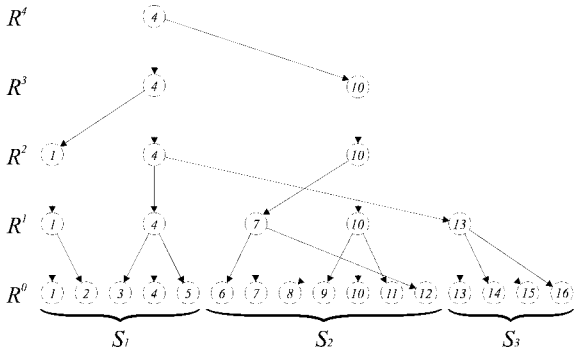


Fig. 8. Tree-structured key frame hierarchy $(L = 4)$.

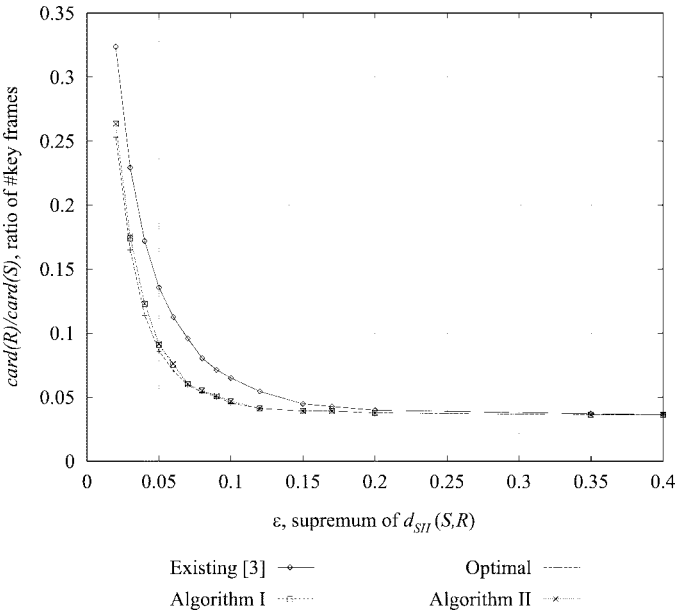


Fig. 9. R-D curve for a movie clip *True Lies.*

δ_i^k cost of the longest path from f_i^k to the leaf nodes in T_i^k, which can be calculated recursively as follows:

$$\delta_i^k = \max_{f_j^{k-1} \in \hat{C}_i^k} (d(f_i^k, f_j^{k-1}) + \delta_j^{k-1}). \tag{15}$$

Now, we describe the following important properties of the tree-structured key frame hierarchy, whose proofs are given in Appendix B.

Property 2: Each internal node (f_i^k) in the tree represents all of its descendants (T_i^k) within the extent of δ_i^k. δ_i^k is bounded by adjustable parameters ϵ_n $(n = 1, 2, \cdots, L)$. In other words, $\delta_i^k \leq \Sigma_{n=1}^{k} \epsilon_n$.

Property 2 means that each level of the key frame hierarchy represents the whole video content at different level of details, making the filtering mechanism in the on-line search phase possible. If a user has a query image and wants to find all the relevant frames in a video sequence for a given distortion ϵ_0, then the DFS with pruning by the value $\delta_i^k + \epsilon_0$ at each node f_i^k can carry out the desired functionality much faster than the serial search, still maintaining the same precision or recall rate. Thus, we can achieve fast access to the frames relevant to a query image in a video sequence or quick rejection for the video sequences whose contents are irrelevant to the query.

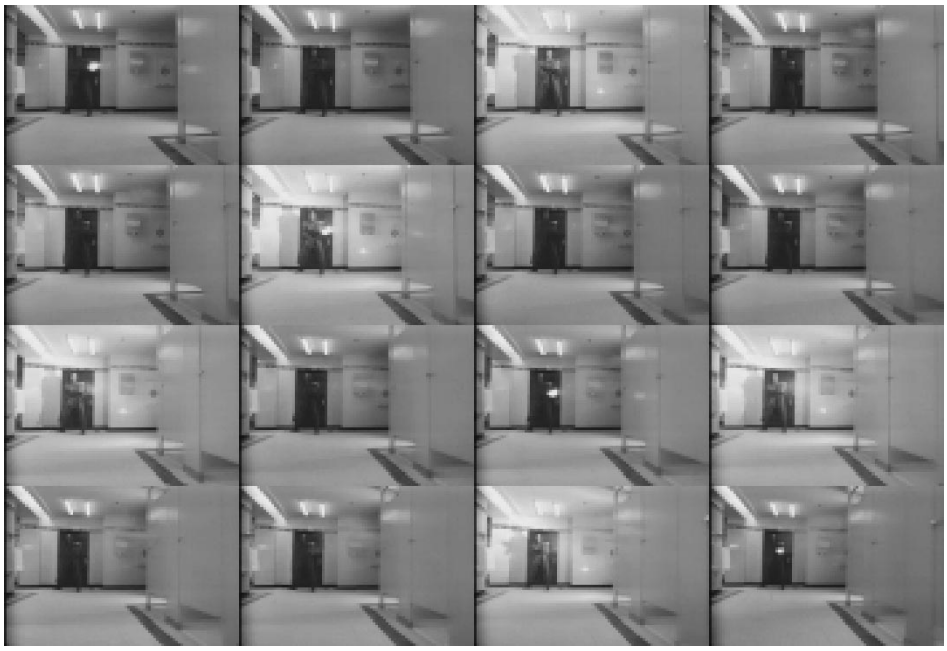


Fig. 10. A shot in *True Lies*. From left to right, top to bottom, each frame is numbered from f_1 to f_{16}.

TABLE III
VARIOUS VIDEO SEQUENCES USED IN EXPERIMENTS

Video sequences	#frames	#shots	min:sec
Commercial	912	16	0:30
Movie Clip	1431	50	1:00
Music Video	6686	163	3:43
News	25974	232	14:26
Sports	27000	181	15:00

The tree-structured key frame hierarchy is a multilevel abstract of a video, which can also facilitate hierarchical browsing similar to [7]. Further, it can be converted to the transition representation such as the scene transition graph [8], in order to provide more friendly browsing to the users, by the analysis of the relationship of the nodes in the tree.

V. EXPERIMENTAL RESULTS

A. Key Frame Extraction at the Shot Level

Experiments are performed on several MPEG-1 video sequences, as listed in Table III. To alleviate both temporal and spatial complexities, we use the dc images without full decompression [13]. We use L_1 norm (7) employing the luminance projection vectors for the dissimilarity measure $d(\cdot)$, as in [3]. In fact, it is important to select an appropriate distance metric. The effectiveness of the KFE depends on the metric used. However, it should be noted that, for a given metric, the proposed approach is (sub)optimal in terms of representation efficiency.

The R-D relation, in which the horizontal axis indicates the maximum of the allowed $d_{\mathrm{SH}}(\cdot)$ (*distortion*) and the vertical axis represents the ratio of the key frames (*rate*), is shown in Fig. 9 to demonstrate the performance. It is analogous to the R-D curve widely used in the area of source coding. From Fig. 9, it is observed that the proposed methods outperform the existing efficient one [3].

In Fig. 10, we show an example of video shot to visually illustrate the superiority of the proposed method. The conventional ones, which are constrained by the sequential order of frames, are apt to select the key frames whenever the luminance condition changes. That is, $R = \{f_1, f_3, f_4, f_6, f_7, f_9, f_{10}, f_{12}, f_{13}, f_{15}, f_{16}\}$. On the other hand, the proposed methods select only two frames (f_3 and f_4); one in light mood and the other in dark mood. It is also experimentally found that the proposed methods work well for various cases. Table IV shows the experimental results for five different types of video sequences. In R-D performance, the proposed methods yield about 30% better results than the existing one for the typical value $\epsilon = 0.04$.

In terms of the extraction time, the proposed algorithms (Algorithm I and II) are more expensive than [3]. Specifically, the method in [3] takes the time of $O(V)$, while Algorithm I takes $O(V^2)$, where V is the number of frames in a video segment of interest. The Algorithm II takes a little more time than Algorithm I. However, the extraction time complexity might not be an issue since in most cases, key frames can be extracted off-line. In this case, due to the better ratio of key frames per fidelity of the proposed approach, the key-frame-based applications such as search and browsing become more efficient in terms of time and space complexities.

B. Structured Index of Video

The content-based search for the query images shown in Fig. 11 is carried out, using the L_1 metric (7) for $d(\cdot)$. Without using any indexing scheme, we should search all of the frames in original video to obtain the results, requiring the image comparisons as many as the number of frames in the video. For example, when an image query is given to *News* video to

(a) (b) (c) (d) (e) (f)

Fig. 11. Query images used. They are (a) *News* first frame, (b) *News* 19 939th frame, (c) *News* 25 974th frame, (d) *Movie clip* 1000th frame, (e) *Music video* 1500th frame, and (f) *Sports* 1000th frame, denoted by Q_1–Q_6.

TABLE IV
RESULTS OF KEY FRAME EXTRACTION FOR FIVE TYPES OF VIDEO SEQUENCES

Video sequences	ϵ	# key frames ($card(R)$)		
		Existing [3]	Algorithm I	Algorithm II
Commercial	0.04	25	19	20
Movie Clip	0.04	243	176	171
Music Video	0.04	1366	999	1032
News	0.04	992	651	648
Sports	0.04	1817	1228	1210

find the similar frames, there would be 25 974 comparisons. The number of comparisons, which is proportional to the online search time, can be greatly reduced by the search scheme using the key frames. The serial search (SS) of all the key frames requires image comparisons as many as $card(R)$. In this scheme, the relevancy between the frame f_i in R and a query q is determined by $d(f_i, q) \leq \epsilon_0$, where ϵ_0 is a user-defined parameter. If we let ϵ_1 be the fidelity constraint used for KFE as in (13), considerable recall rates can be yielded, by setting $\epsilon_0 \geq \epsilon_1$. For the case when the query image exists in the video, it is guaranteed that the key frame that represents the query frame is always retrieved. In the case of the *News* video, the SS of the key frames, extracted by the proposed algorithm (Algorithm I), requires 651 comparisons, which is 40 times faster than the simple SS (1.5 times faster than the SS of the key frames extracted by [3]). More efficient indexing structure, key frame hierarchy, can be constructed using Algorithm I together with four parameters shown in Table V for each video. Table VI shows the number of frames at each level in the key frame hierarchy. For the experiments, we use

$$\delta_i^k = \begin{cases} 0, & \text{if } k = 1 \\ \max\limits_{f_j^{k-1} \in \hat{C}_i^k} (d(f_i^k, f_j^{k-1}) + \delta_j^{k-1}), & \text{if } k = 2, \ldots, L \end{cases} \tag{16}$$

to construct the key frame hierarchy, and the DFS with pruning by the value $\delta_i^k + \epsilon_0$ at each node f_i^k ($\epsilon_0 = \epsilon_1 = 0.04$ is used

Fig. 12. Retrieval results when Q_1 is given to *News* video. *News* (a) first frame(0.000), (b) 13 011st frame (0.016), (c) 3412nd frame (0.019), (d) 9498th frame (0.019), (e) 6466th frame (0.025), and (f) 16 620th frame (0.027). Each number in parenthesis indicates the value of $\hat{d}_1(\cdot)$.

TABLE V
PARAMETERS USED FOR THE CONSTRUCTION OF THE TREE-STRUCTURED KEY FRAME HIERARCHY ($L = 4$)

ϵ_1	ϵ_2	ϵ_3	ϵ_4
0.04	0.06	0.08	0.12

in this experiment) in the on-line search. Although there are some differences depending on the types of video and query used, the tree-structured key frame hierarchy requires two to four times fewer image comparisons than the SS scheme based on the key frames, extracted by Algorithm I (three to six times fewer compared with the SS scheme based on the key frames extracted by [3]), as shown in Table VII.

Fig. 12 shows the retrieval results when Q_1 in Fig. 11 is used as a query image for *News* video. All of them show an anchorman behind the desk. Each of those frames usually corresponds to the start of a new topic, indicating the content of the topic at the right-top position. After these frames are retrieved by the proposed video indexing scheme, a user may be able to rapidly access to the desired news clip.

TABLE VI
NUMBER OF FRAMES AT EACH LEVEL IN THE TREE-STRUCTURED KEY FRAME HIERARCHY

Video sequences	$card(R^1)$	$card(R^2)$	$card(R^3)$	$card(R^4)$
Commercial	19	2	2	2
Movie Clip	176	67	33	6
Music Video	999	481	234	54
News	651	244	91	19
Sports	1228	253	65	12

TABLE VII
RESULTS OF THE RETRIEVAL USING THE TREE-STRUCTURED KEY FRAME HIERARCHY FOR Q_1–Q_6

Video Data	# comparisons					
	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
News	304	351	297	266	125	167
Music Video	244	297	296	192	245	145

Since the mid-1980's, spatial database indexing techniques such as R-tree and R^+-tree have been developed to index multidimensional data [25]. They might achieve performance

similar to the key frame hierarchy proposed in this paper. However, the tree-structured key frame hierarchy is built using the hierarchical application of the KFE algorithms, thus providing a multilevel abstract of a video. Since each level of the key frame hierarchy covers the entire video content at different level of details, it is also expected to facilitate other efficient operations such as browsing.

VI. Conclusion

In this paper, we have proposed an efficient video indexing scheme for the content-based retrieval. The (sub)minimal set of key frames, extracted in combinatorial way, compactly represents the original video with considerable fidelity, thus enabling the fast operations on the video. For the content-based retrieval, more efficient search can be achieved by the hierarchical application of the proposed extraction algorithm. The tree-structured key frame hierarchy, based on the branch-and-bound scheme, greatly reduces the number of image comparisons required for the retrieval. The methods presented in this paper are based on the combinatorial algorithm to enhance the efficiency, but they can also facilitate the sequential operations, such as browsing, by the relatively simple mapping.

All of the key-frame-based approaches represent a video by a small set of representative and decorrelated frame samples. Thus, they lose the continuous motion information of the original video. The proposed algorithms are not the exceptions. By the additional use of some temporal indexing schemes that can utilize the motion information, the proposed approach could be more effective, which is our future research problem.

Appendix A
Proof of Property 1

First assume that $d_{\mathrm{SH}}(S,R)>\epsilon$. Then, by the definition (1), $S \not\subset U(R,\xi), \forall \xi \le \epsilon$. Let us consider only the case of $\xi = \epsilon$. $S \not\subset U(R,\epsilon)$, and it follows that $\exists f \in S$ subject to $f \notin U(R,\epsilon)$. Since $U(R,\epsilon) = \cup_{\hat{f}\in R} B_d(\hat{f},\epsilon)$, $f \notin B_d(\hat{f},\epsilon), \forall \hat{f} \in R$, implying that $d(f,\hat{f})>\epsilon, \forall \hat{f} \in R$ and again $\min_{\hat{f}\in R}\ d(f,\hat{f})>\epsilon$.

Let us now assume that $\exists f \in S$ subject to $\min_{\hat{f}\in R}\ d(f,\hat{f})>\epsilon$, which implies $d(f,\hat{f})>\epsilon, \forall \hat{f} \in R$. The remaining parts of the proof are quite straightforward. $f \notin \{g | d(\hat{f},g) \le \epsilon\}, \forall \hat{f} \in R$. $f \notin \cup_{\hat{f}\in R} B_d(\hat{f},\epsilon) = U(R,\epsilon)$. Since there exists an f satisfying both $f \in S$ and $f \notin U(R,\epsilon)$, $S \not\subset U(R,\epsilon)$. After applying the relation $U(R,\xi_1) \subset U(R,\xi_2)$ whenever $\xi_1 \le \xi_2$, whose proof is omitted here, we conclude that $\forall \xi \le \epsilon$, $S \not\subset U(R,\xi)$. Hence $d_{\mathrm{SH}}(S,R)>\epsilon$.

Appendix B
Proof of Property 2

Note that, by the definition of δ_i^k, the cost of an arbitrary path from f_i^k to a leaf node in T_i^k cannot exceed δ_i^k. Also, the cost of the path is not less than the dissimilarity between two nodes, because of the triangle inequality of $d(\cdot)$. This proves the first property.

The recursive application of the inequality

$$\begin{aligned}\delta_i^k &= \max_{f_j^{k-1}\in\hat{C}_i^k} \left(d(f_i^k, f_j^{k-1}) + \delta_j^{k-1}\right) \\ &\le \max_{f_j^{k-1}\in\hat{C}_i^k} d(f_i^k, f_j^{k-1}) + \max_{f_j^{k-1}\in\hat{C}_i^k} \delta_j^{k-1} \\ &\le d_{\mathrm{SH}}(R^{k-1}, R^k) + \max_{f_l^{k-1}\in R^{k-1}} \delta_l^{k-1},\ \forall i,k \end{aligned} \quad (17)$$

yields $\delta_i^k \le \Sigma_{n=1}^k\ d_{\mathrm{SH}}(R^{n-1},R^n) \le \Sigma_{n=1}^k\ \epsilon_n$.

References

[1] H. S. Chang, S. Sull, and S. U. Lee, "Set-theoretic approach to video key frame extraction," in *Proc. Int. Tech. Conf. Circuits/Systems, Computers and Communications*, Korea, July 1998, vol. 1, pp. 899–902.

[2] F. Arman, R. Depommier, A. Hsu, and M.-Y. Chiu, "Content-based browsing of video sequences," in *Proc. ACM Multimedia'94*, Aug. 1994, pp. 97-103.

[3] M. M. Yeung and B. Liu, "Efficient matching and clustering of video shots," in *Proc. IEEE ICIP'95*, Oct. 1995, vol. 1, pp. 338–341.

[4] R. L. Lagendijk, A. Hanjalic, M. Ceccarelli, M. Soletic, and E. Persoon, "Visual search in a SMASH system," in *Proc. IEEE ICIP'96*, Sept. 1996, vol. 3, pp. 671–674.

[5] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *Proc. IEEE ICIP'98*, Oct. 1998, vol. 1, pp. 866–870.

[6] H. Aoki, S. Shimotsuji, and O. Hori, "A shot classification method of selecting effective key-frames for video browsing," in *Proc. ACM Multimedia'96*, Nov. 1996, pp. 1–10.

[7] D. Zhong, H. Zhang, and S.-F. Chang, "Clustering methods for video browsing and annotation," in *Proc. IS&T/SPIE Storage and Retrieval for Still Image and Video Database IV*, Feb. 1996, vol. 2670, pp. 239–246.

[8] M. M. Yeung, B. Yeo, and B. Liu, "Segmentation of video by clustering and graph analysis," *Comput. Vision Image Understanding*, vol. 71, no. 1, pp. 94–109, July 1998.

[9] Y. Rui, T. S. Huang, and S. Mehrotra, "Exploring video structure beyond the shots," in *Proc. IEEE ICMCS'98*, June 1998, pp. 237–240.

[10] E. Sahouria, "Video indexing based on object motion," Master's thesis, Dept. of Elect. Eng. Comput. Sci., Univ. of California. Berkeley, May 1997.

[11] J. Meng, Y. Juan, and S.-F. Chang, "Scene change detection in a MPEG compressed video sequence," in *Proc. IS&T/SPIE Digital Video Compression: Algorithms and Technologies*, Feb. 1995, vol. 2419, pp. 14–25.

[12] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying scene breaks," in *Proc. ACM Multimedia'95*, Nov. 1995, pp. 189–200.

[13] B. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 533–544, Dec. 1995.

[14] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.

[15] J. R. Munkres, *Topology: A First Course*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[16] M. J. Swain and D. H. Ballard, "Color indexing," *J. Comput. Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[17] B. V. Funt and G. D. Finlayson, "Color constant color indexing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 522–529, May 1995.

[18] I. K. Park, I. D. Yun, and S. U. Lee, "Color image retrieval using hybrid graph representation," *Image Vision Computing*, vol. 17, no. 7, pp. 465–474, May 1999.

[19] J. Huang, S. R. Kumar, M. Mitra, W. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proc. IEEE CVPR'97*, June 1997, pp. 762–768.

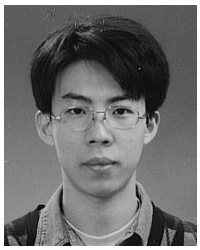
[20] S.-M. Lee, H.-J. Bae, and S.-H. Jung, "Efficient content-based image retrieval methods using color and texture," *ETRI J.*, vol. 20, no. 3, pp. 272–283, Sept. 1998.

[21] J. P. Hayes, *Introduction to Digital Logic Design*. Reading, MA: Addison-Wesley, 1993.

[22] V. Kobla, D. Doermann, and C. Faloutsos, "*VideoTrails*: Representing and visualizing structure in video sequences," in *Proc. ACM Multimedia'97*, Nov. 1997, pp. 335–346.

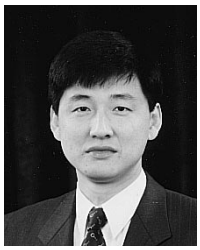
[23] D. DeMenthon, V. Kobla, and D. Doermann, "Video summarization by curve simplification," in *Proc. ACM Multimedia'98*, Sept. 1998, pp. 211–218.

[24] Y. Deng and B. S. Manjunath, "Content-based search of video using color, texture, and motion," in *Proc. IEEE ICIP'97*, Oct. 1997, vol. 2, pp. 534–537.
[25] E. Bertino, B. C. Ooi, R. Sacks-Davis, K.-L. Tan, J. Zobel, B. Shidlovsky, and B. Catania, *Indexing Techniques for Advanced Database Systems*. Norwell, MA: Kluwer, 1997.
[26] S. W. Smoliar and H. Zhang, "Content-based video indexing and retrieval," *IEEE Multimedia Mag.*, vol. 1, no. 2, pp. 62–72, 1994.
[27] H. Zhang, J. Y. A. Wang, and Y. Altunbasak, "Content-based video retrieval and compression: A unified solution," in *Proc. IEEE ICIP'97*, Oct. 1997, vol. 1, pp. 13–16.
[28] Y. Nakamura and T. Kanade, "Semantic analysis for video contents extraction—Spotting by association in news video," in *Proc. ACM Multimedia'97*, Nov. 1997, pp. 393–401.
[29] "MPEG-7: Context, objectives and technical roadmap, V.11," ISO/IEC JTC1/SC29/WG11 N2729, Seoul, Korea, Mar. 1999.



Hyun Sung Chang (M'99) received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, Korea, in 1997 and 1999, respectively.

In March 1999, he joined the Radio & Broadcasting Technology Laboratory, Electronics and Telecommunications Research Institute, Taejon, Korea, as a Member of Engineering Staff. His research interests include content-based video analysis and representation, image understanding, and other issues on image and video technologies.



Sanghoon Sull (S'79–M'81) received the B.S. degree (with honors) in electronics engineering from the Seoul National University, Korea, in 1981, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology in 1983, and the Ph.D. degree in electrical and computer engineering from the University of Illinois, Urbana-Champaign, in 1993.

In 1983–1986, he was with the Korea Broadcasting Systems, working on the development of the teletext system. In 1994–1996, he conducted research on motion analysis at the NASA Ames Research Center. In 1996–1997, he conducted research on video indexing/browsing and was involved in the development of the IBM DB2 Video Extender at the IBM Almaden Research Center. He joined the School of Electrical Engineering at the Korea University as an Assistant Professor in 1997 and is now an Associate Professor. His current research interests include multimedia data management including search/browsing, MPEG-7, image processing, and Internet applications.

Sang Uk Lee (S'75–M'80–SM'99) received the B.S. degree from Seoul National University, Seoul, Korea, in 1973, the M.S. degree from Iowa State University, Ames, in 1976, and the Ph.D. degree from the University of Southern California, Los Angeles, in 1980, all in electrical engineering.

In 1980–1981, he was with General Electric Co., Lynchburg, VA, working on the development of digital mobile radio. In 1981–1983, he was a Member of Technical Staff, M/A-COM Research Center, Rockville, MD. In 1983, he joined the Department of Control and Instrumentation Engineering at Seoul National University as an Assistant Professor, where he is now a Professor of the School of Electrical Engineering. Currently, he is also affiliated with the Automation and Systems Research Institute and the Institute of New Media and Communications at Seoul National University. His current research interests are in the areas of image and video signal processing, digital communication, and computer vision. He was an Editor-in-Chief for the *Transaction of the Korean Institute of Communication Science* from 1994 to 1996. Currently, he is a member of the editorial board of the *Journal of Visual Communication and Image Representation*.

Prof. Lee is a member of Phi Kappa Phi. He is an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.