EFFICIENT VIDEO SIMILARITY MEASUREMENT WITH VIDEO SIGNATURE

Sen-ching S. Cheung and Avideh Zakhor

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720
{cheungsc, avz}@eecs.berkeley.edu

ABSTRACT

The Video Signature method has been proposed in [1, 2, 3] as a technique to efficiently summarize video for visual similarity measurements. In this paper, we develop the necessary theoretical framework to analyze this method. We define our target video similarity measure based on the fraction of similar clusters shared between two video sequences. This measure is too computationally complex to be deployed in database applications. By considering this measure geometrically on the image feature space, we find that it can be approximated by the volume of the intersection between Voronoi cells of similar clusters. In the Video Signature method, sampling is used to estimate this volume. By choosing an appropriate distribution to generate samples, and ranking the samples based upon their distances to the boundary between Voronoi cells, we demonstrate that our target measure can be well approximated by the Video Signature method. Experimental results on a large dataset of web video and a set of MPEG-7 test sequences with artificially generated similar versions are used to demonstrate the retrieval performance of our proposed techniques.

1. INTRODUCTION

The proliferation of video content on the web makes similarity detection an indispensable tool in web data management, searching, and navigation. In [1, 2, 3], we have proposed a randomized algorithm, called the Video Signature (ViSig) method, to identify highly similar video sequences in large databases. In this paper, we further develop the necessary theoretical framework to interpret and enhance our algorithms. This paper is organized as follows: in Section 2, we define our target similarity measure, called the Ideal Video Similarity (IVS), and explain how we use the ViSig method to estimate IVS. In Section 3, we analyze the scenarios where IVS cannot be reliably estimated by ViSig, and propose a number of heuristics to improve the estimation. Experimental results are presented in Section 4 to demonstrate the retrieval performance of our proposed methods. Due to space limitation, proofs to various propositions in this paper are not included, and interested readers are referred to [4] for details.

2. IDEAL VIDEO SIMILARITY AND VIDEO SIGNATURE

This section defines the video similarity model used in this paper, and describes how it can be efficiently estimated by the ViSig method. We assume that individual frames in a video are represented by high-dimensional feature vectors from a metric space $(F, d(\cdot,\cdot))$. In order to be robust against editing changes in temporal domain, we define a video to be a finite set of feature vectors and ignore any temporal ordering. The metric $d(x, y)$ measures the visual dissimilarity between frames x and y. We assume that frames x and y are visually similar to each other if and only if $d(x, y) \leq \epsilon$ for an $\epsilon > 0$ independent of x and y. Given a video X, we define its clustering $[X]_\epsilon$ to be a collection of non-intersecting subsets of X such that if $d(x_i, x_j) \leq \epsilon$, x_i and x_j must belong to the same subset or cluster in $[X]_\epsilon$. Such a clustering can be easily computed using the single-link algorithm [6]. When defining the similarity between two video sequences X and Y, we consider the clustered union $[X \cup Y]_\epsilon$. We call the clusters in $[X \cup Y]_\epsilon$ Similar Clusters if they contains frames from both sequences, and define IVS as the percentage of Similar Clusters in $[X \cup Y]_\epsilon$:

$$\mathrm{ivs}(X, Y; \epsilon) \stackrel{\Delta}{=} \Big(\sum_{C \in [X \cup Y]_\epsilon} 1_{C \cap X} \cdot 1_{C \cap Y} \Big) / |[X \cup Y]_\epsilon| \qquad (1)$$

where 1_A is the indicator function and $|\cdot|$ denotes the cardinality of a set. It is complex to precisely compute IVS. The clustering used in IVS depends on the distances between frames from the two sequences. Thus, we need to compute all pairwise distances before running the clustering algorithm. In addition, the computation requires the entire video to be stored. The complex computation and large storage requirements are clearly undesirable for large database applications. The main theme of this paper is to develop efficient algorithms in estimating IVS.

Since our overall goal is to detect "near duplicate" video sequences, the exact similarity value is not required. Sampling techniques can thus be used to estimate IVS. Specifically, two video sequences are declared to be similar if there are sufficiently large number of "similar" sampled frames between them. The naive scheme of sampling frames independently from two sequences, however, does not work. Even when the IVS between two sequences is one, it can be shown that an average of $\sqrt{l}$ sampled frames are needed to find just one pair of similar frames, where l denotes the number of frames in each sequence [4]. This is due to the small probability of finding a pair of similar frames between two independent sets of sampled frames. Instead of independent sampling, the ViSig method introduces dependence by selecting frames in each video that are similar to a set of predefined random feature vectors. As a result, the probability in finding similar frames from two video clips with IVS of one is much higher. The number of pairs of similar frames found by the ViSig method strongly depends on the IVS, but does not have a one-to-one relationship with it. We call the form of similarity estimated by the ViSig method the Voronoi Video Similarity (VVS), as described below.

The term "Voronoi" in VVS is borrowed from a geometrical concept called the Voronoi Diagram. Given a l-frame video $X = \{x_t : t = 1, \ldots, l\}$, the Voronoi Diagram $V(X)$ of X is a partition of the feature space F into l Voronoi Cells $V_X(x_t)$. By definition, the Voronoi Cell $V_X(x_t)$ contains all the vectors in F closer to $x_t \in X$ than to any other frames in X, i.e. $V_X(x_t) \stackrel{\Delta}{=} \{s \in F : g_X(s) = x_t \text{ and } x_t \in X\}$, where $g_X(s)$ denotes the frame in X closest to s. We can extend the idea of the

This work was sponsored by NSF Grant CCR-9903368 and AFOSR Award FDF49620-99-1-0062.

Voronoi Diagram to video clusters by merging Voronoi Cells of all the frames belonging to the same cluster. Given two video sequences X and Y and their corresponding Voronoi Diagrams, we define the Similar Voronoi Region (SVR), denoted as $R(X,Y;\epsilon)$, to be the union of all the intersection between the Voronoi Cells $V_X(x)$ and $V_Y(y)$ with $d(x,y) \leq \epsilon$:

$$R(X,Y;\epsilon) \triangleq \bigcup_{d(x,y)\leq\epsilon} V_X(x) \cap V_Y(y). \tag{2}$$

It is easy to see that SVR is large if video X and Y have a large number of frames that are similar to each other. Thus, we define the VVS between X and Y, $\mathrm{vvs}(X,Y;\epsilon)$, to be the volume of the corresponding SVR:

$$\mathrm{vvs}(X,Y;\epsilon) \triangleq \sum_{d(x,y)\leq\epsilon} \mathrm{Vol}(V_X(x) \cap V_Y(y)). \tag{3}$$

The volume function $\mathrm{Vol} : \Omega \rightarrow \mathbb{R}$ is the Lebesgue measure over the set, Ω, of all the measurable subsets in F. F is assumed to be compact with $\mathrm{Vol}(F) = 1$. A simple pictorial example to demonstrate the use of VVS is shown in Figure 1(a). The feature space is represented as a 2D square. Dots and crosses signify frames from two different video sequences, whose Voronoi Cells are separated by solid and broken lines respectively. Frames closer than ϵ are connected by dotted lines. The shaded region represents the SVR. The VVS, which measures the area of the shaded region, is about 1/3, the same as the IVS in this case.

It is straightforward to estimate VVS by random sampling: we first generate a set S of m independent uniformly-distributed Seed Vectors (SV) $s_1, \ldots, s_m$. We can then estimate $\mathrm{vvs}(X,Y;\epsilon)$ based on the percentage of SV's that are inside $R(X,Y;\epsilon)$. To determine if a particular SV s falls inside $R(X,Y;\epsilon)$, we only need to check whether $g_X(s)$, the frame closest to s in X, and $g_Y(s)$ are within ϵ of each other. In other words, we can estimate VVS between X and Y by using only the m-frame tuple $\overrightarrow{X_S} \triangleq (g_X(s_1), \ldots, g_X(s_m))$ from video X and $\overrightarrow{Y_S}$ from Y. We call $\overrightarrow{X_S}$ the signature of X with respect to S. We define Basic ViSig Similarity (VSS_b) between $\overrightarrow{X_S}$ and $\overrightarrow{Y_S}$ to be the percentage of $g_X(s)$ and $g_Y(s)$ that are within ϵ from each other:

$$\mathrm{vss}_b(\overrightarrow{X_S}, \overrightarrow{Y_S}; \epsilon, m) \triangleq (\sum_{i=1}^{m} 1_{\{d(g_X(s_i), g_Y(s_i))\leq\epsilon\}})/m \tag{4}$$

$\mathrm{vss}_b(\overrightarrow{X_S}, \overrightarrow{Y_S}; \epsilon, m)$ forms an unbiased estimate of the VVS between X and Y. We refer to this approach of generating signatures and computing VSS_b as the Basic ViSig method. In order to apply the Basic ViSig method to a large number of video sequences, we must use the same SV set to generate all the signatures before we can compute VSS_b between an arbitrary pair of video sequences.

3. DIFFERENCES BETWEEN IVS AND VVS

We have shown in the previous section that the VVS between two video sequences can be efficiently estimated by the Basic ViSig method. Unfortunately, the estimated VVS does not necessarily reflect the target measure of IVS as defined in Equation (1). Two problems can arise: first, VVS can be either larger or smaller than IVS because of the non-uniform sizes of the Voronoi Cells. Second, VVS can be smaller than IVS because of the presence of a special region in the Voronoi Diagram called the Voronoi Gap (VG). In this section, we explain these two problems and propose heuristic solutions to improve the estimation.

3.1. Seed Vector Generation

When all clusters in the clustered union between two video sequences clump together in a small area, some Voronoi Cells can be significantly larger than the others. If the Similar Clusters happen to reside in the smaller Voronoi Cells, the VVS is smaller than the IVS. On the other hand, if the Similar Clusters are in the larger Voronoi Cells, the VVS becomes much larger. As the Basic ViSig method estimates IVS based on uniformly-distributed SV's, the variation in the sizes of Voronoi Cells affects the accuracy of the estimation. One possible method to amend the Basic ViSig method is to generate SV's based on a probability distribution such that the probability of a SV being in a Voronoi Cell is independent of the size of the Cell. Specifically, for two sequences X and Y, we can define the Probability Density Function (PDF) at the feature vector u based on the distribution of Voronoi Cells in $[X \cup Y]_\epsilon$ as follows:

$$f(u; X \cup Y) \triangleq 1/[|[X \cup Y]_\epsilon| \cdot \mathrm{Vol}(V_{X\cup Y}(C))] \tag{5}$$

where C is the cluster in $[X \cup Y]_\epsilon$ with $u \in V_{X\cup Y}(C)$. With this PDF, the probability of a random vector u being inside the Voronoi Cell $V_X(C)$ for an arbitrary cluster $C \in [X \cup Y]_\epsilon$ is given by $\int_{V_X(C)} f(u; X \cup Y)\, du = 1/|[X \cup Y]_\epsilon|$. This probability does not depend on C, and thus, it is equally likely for u to be inside the Voronoi Cell of any cluster in $[X \cup Y]_\epsilon$. Recall that if we use uniform distribution to generate random SV's, VSS_b forms an unbiased estimate of the VVS defined in Equation (3). If we use $f(u; X \cup Y)$ to generate SV's instead, VSS_b becomes an estimate of the following general form of VVS:

$$\sum_{d(x,y)\leq\epsilon} \int_{V_X(x)\cap V_Y(y)} f(u; X \cup Y)\, du. \tag{6}$$

Equation (6) reduces to Equation (3) when $f(u; X \cup Y)$ is replaced by the uniform distribution, i.e. $f(u; X \cup Y) = 1$. As shown by the following proposition, this general form of VVS is equivalent to IVS under certain conditions.

Proposition 3.1 *Given two video sequences X and Y, assume clusters in $[X]_\epsilon$ and $[Y]_\epsilon$ are either identical, or share no frames that are within ϵ of each other. Then, $\mathrm{ivs}(X,Y;\epsilon)$ is equivalent to the general form of VVS given by Equation (6).*

The significance of this proposition is that if we can generate SV's with $f(u; X \cup Y)$, it is possible to estimate IVS using the same tool as to estimate VVS, namely VSS_b. The condition that all clusters in X are Y are either identical or far away from each other is to avoid the formation of VG, which is expounded in Section 3.2.

In practice, it is impossible to use $f(u; X \cup Y)$ to generate SV's. This is because $f(u; X \cup Y)$ is specific to the two sequences being compared, while the Basic ViSig method requires the same set of SV's to be used by all sequences in the database. A heuristic approach for SV generation is to first select a set Ψ of training video sequences that resemble video sequences in the target database. Denote $T \triangleq \bigcup_{Z\in\Psi} Z$. We can then generate SV based on the PDF $f(u;T)$, which ideally resembles the target $f(u; X \cup Y)$ for an arbitrary pair of X and Y in the database. To generate a random SV s based on $f(u;T)$, we follow a four-step SV generation method as follows: Given a particular value of ϵ_{sv}, we first identify all the clusters in $[T]_{\epsilon_{sv}}$ using the single-link algorithm. Second, as $f(u;T)$ assigns equal probability to the Voronoi Cell of each cluster in $[T]_{\epsilon_{sv}}$, a cluster C' is randomly selected from $[T]_{\epsilon_{sv}}$ so that we can generate the SV s within $V_T(C')$. As $f(u;T)$ is constant over $V_T(C')$, we should ideally generate s as a uniformly-distributed random vector over $V_T(C')$. One possible way is to repeatedly generate uniform sample vectors over the entire feature space until a vector is found inside $V_T(C')$. This procedure may take an exceedingly long time if $V_T(C')$ is small. To simplify the generation, we select one of the frames in C' at random and output it as the next SV. Finally, we repeat the above process until the required number of SV's have been selected. In Section 4, we compare retrieval performances on real video sequences based on SV's generated using this algorithm and the uniform distribution.

3.2. Voronoi Gap (VG)

We have shown in Proposition 3.1 that the general form of VVS using an appropriate PDF is identical to the IVS, provided that all clusters between the two sequences are either identical or far away from each other. As feature vectors are not perfect in modeling human visual system, visually similar clusters may result in feature vectors that are close but not identical to each other. Let us consider the example in Figure 1(b) where frames from two sequences are not identical but within ϵ from each other. Clearly, the IVS is one. Consider the Voronoi Diagrams of the two sequences. Since the boundaries of the two Voronoi Diagrams do not exactly coincide with each other, the SVR, as indicated by the shaded area, does not occupy the entire feature space. As the general form of VVS defined in Equation (6) is the weighted volume of the SVR, it is strictly less than the IVS. The difference between the two similarities is due to the unshaded region in Figure 1(b). If a SV s falls within the unshaded region in Figure 1(b), we can make two observations about the corresponding signature frames $g_X(s)$ and $g_Y(s)$: (1) they are far apart from each other, i.e. $d(g_X(s), g_Y(s)) > \epsilon$; (2) they both have similar frames in the other video, i.e. there exists $x \in X$ and $y \in Y$ such that $d(x, g_X(s)) \le \epsilon$ and $d(y, g_Y(s)) \le \epsilon$. These observations define a unique characteristics of the unshaded region, which we refer to as the Voronoi Gap (VG) and denote as $G(X, Y; \epsilon)$. Any SV in the VG between two sequences produces a pair of dissimilar signature Frames, even though both signature frames have a similar match in the other video.

The example in Figure 1(b) seems to suggest that VG is small if ϵ is small. Nonetheless, in some particular feature spaces such as the hamming cube, the volume of VG can be quite significant even for small ϵ [4]. As such, it is important to identify those SV's that are inside the VG and discard the corresponding signature frames in the estimation of IVS. In Figure 1(b), we observe that for an arbitrary vector s in VG, each sequence has a pair of dissimilar frames that are roughly equidistant to s: x and $g_X(s)$ in the "dot" sequence, and y and $g_Y(s)$ in the "cross" sequence. This "equidistant" condition is refined in the following proposition to upper-bound the difference between distance of s and x, and distance of s and $g_X(s)$ by 2ϵ:

Proposition 3.2 *Let X and Y be two video sequences. Assume all the frames in each cluster of $[X \cup Y]_\epsilon$ are within ϵ from each other. For any SV $s \in G(X, Y; \epsilon)$, there exists a frame $x \in X$ with $d(x, g_X(s)) > \epsilon$ such that $d(x, s) - d(g_X(s), s) \le 2\epsilon$.*

We can use the converse of Proposition 3.2 to test if a SV s can ever be inside a VG of X: define a Ranking Function $Q(\cdot)$ for the signature Frame $g_X(s)$,

$$Q(g_X(s)) \triangleq \min_{x \in X,\, d(x, g_X(s)) > \epsilon} d(x, s) - d(g_X(s), s). \qquad (7)$$

Based on Proposition 3.2, if $Q(g_X(s)) > 2\epsilon$, s cannot be inside the VG formed between X and any other sequence. In practice, however, this condition might be too restrictive in that it might not allow us to find any SV. Recall that Proposition 3.2 only provides a sufficient and not a necessary condition for a SV to be in VG. Thus, even if $Q(g_X(s)) \le 2\epsilon$, it does not necessarily imply that s will be inside the VG between X and any particular sequence. Intuitively, in order to minimize the chance of being inside any VG, it makes sense to use a SV s with as large of a $Q(g_X(s))$ value as possible. Thus, we can generate a large number of signature frames and only use those with the largest ranking function values for similarity measurements. Let $m' > m$ be the number of frames in each signature. After we generate the signature $\overrightarrow{X_S}$ with respect to a set S of m' SV's, we compute and rank $Q(g_X(s))$ for all $g_X(s)$ in $\overrightarrow{X_S}$. Analogous to VSS$_b$ defined in Equation (4), we define the Ranked ViSig Similarity (VSS$_r$) based on the top-$m/2$ ranked signature frames:

$$\mathrm{vss}_r(\overrightarrow{X_S}, \overrightarrow{Y_S}; \epsilon, m) \triangleq \frac{1}{m}\sum_{i=1}^{m/2}[1_{\{d(g_X(s_{j[i]}), g_Y(s_{j[i]})) \le \epsilon\}} + 1_{\{d(g_X(s_{k[i]}), g_Y(s_{k[i]})) \le \epsilon\}}] \qquad (8)$$

where $j[1], \ldots, j[m']$ and $k[1], \ldots, k[m']$'s denote the relative rankings of the signature Frames in $\overrightarrow{X_S}$ and $\overrightarrow{Y_S}$ respectively, i.e. $Q(g_X(s_{j[1]})) \ge \ldots \ge Q(g_X(s_{j[m']}))$ and $Q(g_Y(s_{k[1]})) \ge \ldots \ge Q(g_Y(s_{k[m']}))$. We call this method of generating signatures and computing VSS$_r$ the Ranked ViSig method. Notice in the right hand side of Equation (8), the first term uses the top-$m/2$ ranked signature frames from $\overrightarrow{X_S}$ to compare with the corresponding signature frames in $\overrightarrow{Y_S}$, and the second term uses the top-$m/2$ ranked frames from $\overrightarrow{Y_S}$. Computing VSS$_r$ thus requires m metric computations, the same as VSS$_b$. This provides an equal footing in complexity to compare the retrieval performances between these two methods in Section 4.

4. EXPERIMENTAL RESULTS

In this section, we present experimental results to demonstrate the performance of the ViSig methods. Four 178-bin HSV color histograms, each representing a quadrant of a frame, is used as the feature vector in all our experiments. We compare color histograms with both l_1 metric and a modified l_1 distance, which discards the dominant color from the measurement [4].

Two sets of data are used in our experiments. The first dataset consists of 15 video sequences selected from the MPEG-7 video CD set: v1, v3, v4, v5,v6, v7, v8, and v9 [5]. The average length of the test sequences is 30 minutes. We perform two experiments with this dataset to verify the heuristics proposed in Section 3. In the first experiment, we demonstrate the effect of the choice of SV's in the ViSig methods on approximating IVS. We randomly drop frames from each sequence to artificially create similar versions at different levels of IVS. Signatures with respect to two different sets of SV's are created for all the sequences and their similar versions. The first set of SV's are independent random vectors, uniformly distributed on the high-dimensional histogram space. We use the seed generation method described in Section 3.1 with $\epsilon_{sv} = 2.0$ to generate the second set of SV's. The training set consists of 4000 random images from the Corel Stock Photo Collection. We use 100 SV's to generate each signature, and measure the VSS$_b$ between all sequences and their similar versions at IVS levels of 0.8, 0.6, 0.4 and 0.2. Table 1 shows the averages and standard deviations of the VSS$_b$ values over all the test sequences. As expected, the VSS$_b$ based on Corel images are closer to the underlying IVS than those based on random vectors. In addition, the fluctuations in the estimates are far smaller with the Corel images. The experiment thus shows that it is advantageous to use SV's that approximate the feature vector distribution of the target data.

In the second experiment, we compare the Basic ViSig method with the Ranked ViSig method in identifying sequences with IVS of one, under small feature vector displacements. In this experiment, we create similar video by adding noise to individual frames. Most of the real-life noise processes such as compression are highly video dependent, and cannot provide a wide-range of controlled noise levels for our experiment. As such, we introduce noise at different ϵ levels by randomly changing the colors of different percentages of pixels in each frame. Five ϵ levels are tested in our experiments: 0.2, 0.4, 0.8, 1.2 and 1.6. After injecting noise to create the similar video, a 100-frame Basic Signature $(m = 100)$ and a 500-frame Ranked Signature $(m' = 500)$ are generated for each video. All SV's are randomly sampled from the Corel dataset. To ensure the same computational complexity between the two methods, the top 50-ranked signature frames are

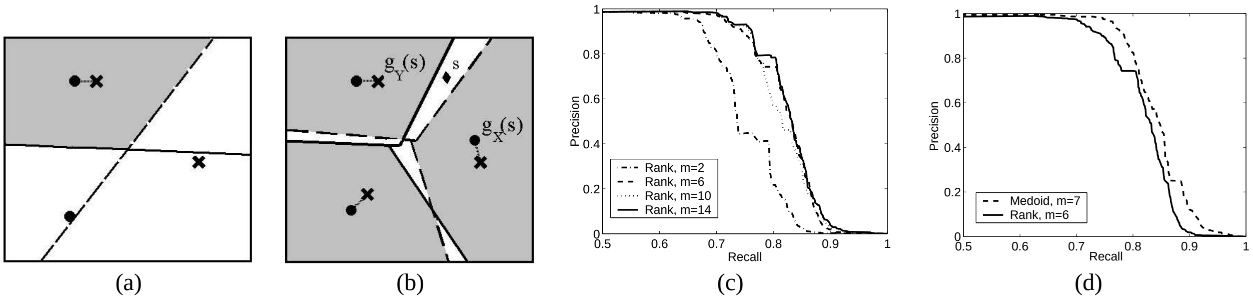

Fig. 1. *(a) The shaded area, normalized by the area of the entire space, is equal to the VVS. (b) The unshaded area represents the Voronoi Gap. (c) Precision-Recall graphs for the Ranked ViSig methods with* $m = 2, 6, 10, 14$*. (d) Precision-Recall graphs for the Ranked ViSig method with* $m = 6$ *(solid) and k-medoid with 7 representative frames (broken).*

used in computing VSS_r. The averages and standard deviations of VSS_b's and VSS_r's over all test sequences are shown in Table 2. Since the IVS is fixed at one, the closer the average similarity is to one, the better the approximation is. Even though both methods show degradation as the noise level increases, as expected, VSS_r measurements are much closer to one than VSS_b.

We also use a much larger dataset to demonstrate how the Ranked ViSig method can be applied in a realistic application. The dataset consists of 46,356 video sequences, crawled from the web between August and December in 1999. We test our algorithms based on their retrieval performance using a manually-derived ground-truth set, which consists of 443 video sequences in 107 clusters [3]. We declare two video sequences X and Y to be similar if $\mathrm{vss}_r(\overrightarrow{X_S}, \overrightarrow{Y_S}; \epsilon, m) \leq 0.5$. A spectrum of different recall and precision values are measured by varying ϵ. SV's are randomly selected by the seed vector generation method of Section 3.1, with ϵ_{sv} set to 2.0, from a set of keyframes representing the video sequences in the dataset. A 100-frame signature $(m' = 100)$ is generated for each video and we measure the precision and recall for four different m values: 2, 6, 10 and 14. The resultant plot is shown in Figure 1(c). There is a substantial gain in performance when m increases from two to six. Further increase in m does not produce any significant gain. All the precision-recall curves decline sharply once they reach beyond 75% recall and 90% precision. Thus, $m = 6$ is adequate in retrieving our ground-truth from the dataset. We also compare the retrieval performance between the Ranked ViSig method and another summarization method called the k-medoid method[8, 9]. The k-medoid method represents each video by k of its frames, or medoids, which minimize the sum of distances between the medoids and all other frames in the video. The method of computing the k-medoid representation is significantly more complex than the Ranked ViSig method: for a video with l frames, the k-medoid algorithm is iterative with each iteration running at $O(l^2)$ time [8]. On the other hand, the Ranked ViSig method is a single-pass linear algorithm. We plot the precision-recall curves for the k-medoid method with $k = 7$ and the six-frame Ranked ViSig method in Figure 1(d). The k-medoid technique provides a slightly better retrieval performance. The advantage seems to be small considering the complexity advantage of the ViSig method.

5. CONCLUSION

In this paper, we have proposed the ViSig method which summarizes a video sequence by extracting the frames closest to a set of randomly selected SV's called signatures. By comparing the signature frames between two video sequences, we obtain an unbiased estimate of their VVS. In order to reconcile the difference between VVS and IVS, the SV's used must resemble the frame statistics of video in the target database. In addition, we propose a ranking method to identify those SV's that are least likely to be inside the VG. Experimental results on a large dataset of web video and a set of MPEG-7 test sequences are presented to demonstrate the retrieval performance of our proposed techniques.

6. REFERENCES

[1] S.-C. Cheung, A. Zakhor, “Estimation of web video multiplicity,” *Proc. SPIE*, vol. 3964, p. 34–6, Jan. 2000.

[2] S.-C. Cheung, A. Zakhor, “Efficient video similarity measurement and search,” *ICIP 2000*, vol. I, p. 85–9, Sept. 2000.

[3] S.-C. Cheung, A. Zakhor, “Video similarity detection with video signature clustering,” *ICIP 2001*, vol. I, p. 649–52, Sept. 2001.

[4] S.-C. Cheung, A. Zakhor, “Efficient Video Similarity Measurement with Video Signature,” Submitted to IEEE Trans. on CSVT, Jan., 2002.

[5] MPEG-7 Requirement Group, “Description of MPEG-7 Content Set,” ISO/IEC JTC1/SC29/WG11/N2467, 1998.

[6] R. Sibon, “Slink: An optimally efficient algorithm for the single-link cluster method,” The Computer Journal, vol. 16, no. 1, p. 30–4, 1973.

[7] H.L. Royden, “Real Analysis,” Macmillan Publishing Company, 1988.

[8] L. Kaufman, P.J. Rousseeuw, “Finding Groups in Data,” Johh Wiley & Sons, New York, 1990.

[9] H.S. Chan, S. Sull, S.U. Lee, “Efficient Video Indexing Scheme for Content-Based Retrieval,” IEEE Trans. on CSVT., vol. 9, no. 8, p. 1269–79, 1999.

Seed Vectors	Uniform				Non-uniform based on Corel Images			
IVS	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2
Average	0.873	0.499	0.429	0.166	0.828	0.581	0.428	0.187
Stddev	0.146	0.281	0.306	0.169	0.060	0.083	0.051	0.046

Table 1. *Comparison between using uniform random and corel image SV's.*

Algorithm	VSS_b					VSS_r				
ϵ	0.2	0.4	0.8	1.2	1.6	0.2	0.4	0.8	1.2	1.6
Average	0.879	0.761	0.628	0.518	0.424	1.000	0.998	0.933	0.837	0.744
Stddev	0.038	0.061	0.061	0.080	0.082	0.000	0.006	0.052	0.070	0.088

Table 2. *Comparison between using the Basic and the Ranked ViSig method under different levels of perturbation.*