Efficient Video Multiplicity Measurement And Search

Qualifying Examination Proposal (May, 2001)
Sen-ching Samson Cheung, cheungsc@eecs

1 Introduction

The amount of information on the world wide web has grown enormously since its creation in 1990. Since there is no central management of information on the web, finding duplicate content is inevitable. Overly-duplicated contents increase the effort in information mining for both human and artificial agents. This problem is in fact quite severe: as reported by Shivakumar and Garcia-Molina[1] in 1998, around 46% of all the text documents on the web have at least one "near-duplicate" – a document that is identical except for low level details such as formatting. Multimedia content, particularly video content, is likely to be more problematic than text documents. This can be attributed to the fact that video content is often mirrored in multiple locations, formats and bitrates to facilitate downloading. Multimedia authoring tools also enable users to slightly modify existing video content and republish them on the web. Identifying all similar contents on the web can be beneficial to many web retrieval applications. Specifically:

1. Search results can be clustered to allow easy browsing.
2. During network outages or in cases of expired links, an alternative copy in a different location can provide fault tolerance.
3. Without using costly transcoding procedures, the search engine can present the best version to users based on resource/location estimation and users' specifications. The simplest example is to choose the copy which is physically closest to the user.
4. Clustering of information provides useful cues for web data mining. For example, the inclusion of the similar content in two different users' homepages is a strong indication of the two users belonging to the same community[2].

In this thesis work, I develop efficient algorithms to both measure visual similarity between video sequences, and to search for similar video content in large databases such as the web. I define similar video sequences to be those with roughly the same content but possibly compressed at different qualities and formats, or undergone minor editing in spatial and temporal domains. Such editing effects include cropping, addition of logos, reordering of shots and different transition effects, etc. In order to compactly represent video sequences for comparison, I develop a novel linear-time randomized algorithm to summarize a video sequence into a *video signature*[3]. A video signature is a set of high-dimensional feature vectors representing a small number of specially-selected frames from the video sequence. I illustrate in my experiments that using video signatures can reliably measure video similarity. It is well-known that performing efficient similarity search on a large database of high-dimensional vectors is a difficult task[4]. In this regard, I apply dimension reduction techniques to significantly reduce the complexity in finding similar video signatures in large databases[5]. To further

enhance retrieval performance, I also develop a novel graph-theoretical clustering algorithm to uncover structures from a large database of video signatures[6]. In order to test my proposed algorithms, I devise a web crawler to gather video clips from the web. Between September and November of 1999, I collected over 46,000 video clips totaling approximately 1,800 hours of data[3]. Based on this dataset, I implement a prototype text-based search engine for the video clusters. The search engine can be accessed at http://www-video.eecs.berkeley.edu/~cheungsc/cluster/search.html.

This paper is organized as follows: after reviewing some background and related work in section 2, I describe the video signature algorithm and present some experimental results in Section 3. The dimension reduction scheme for video signatures is described in Section 4, and the clustering algorithm in Section 5. Finally, I conclude this paper in Section 6 by summarizing the contributions.

2 Background

2.1 Video similarity measurement

In recent years, there has been a significant amount of research on similarity measurement and search in video databases. Perry et al. provide an excellent review of this area in [7]. Most existing work in this area focuses on developing video processing techniques to match our intuitive notion of similarity. Their experimental results are based on either high-quality, domain-specific video databases[8, 9, 10], or a small set of video clips from the web[11, 12]. In contrast, I focus on a much larger set of web video clips. These test data are extremely diverse in both content and quality, and thus are not amenable to domain specific techniques.

In general, measuring similarity between two video sequences is computationally intensive. In [8, 9, 13], the video similarity is based on computing the warping distance, and the computational complexity is proportional to the length of the longer video. Hausdorff metric is used in [14] and the complexity is proportional to the product of the lengths. If the primary concern is to identify approximately equivalent video sequences, complexity can be reduced by focusing on a few distinctive features, or using sampling techniques. For example, Indyk et al.[12] use the time series of shot-change durations as the signature for a video sequence. However, this method is not applicable to general web video clips as the majority of them are short or contain very few shot changes. Another approach to reduce complexity is to only compare the similarity between small sets of significant frames, called the keyframes, extracted from the two sequences. Since most keyframe-selection algorithms in the literature are designed to facilitate browsing and story-boarding[15, 16, 17], they usually produce too many keyframes and provide no performance guarantee for estimating the underlying video similarity. In [14], Chang et al. formulate the selection of keyframes as an optimization problem in which the Hausdorff distance between the keyframes and the original video sequence is minimized. Given a fixed number of keyframes, the (Hausdorff) distance between two such sets of optimal keyframes provides the best estimate of the distance between the two underlying video sequences. Chang et al. show that such an optimization problem is NP-complete. They approximate the solution by an iterative algorithm in which each step is $O(T^2)$ complex, with T being the length of the video. Such an algorithm is too complex to be applied in practice. If small classification error can be tolerated, it is possible to design much faster methods in generating keyframes. My proposed video signature scheme can provide a single-pass linear-time algorithm to generate keyframes which can be used to reliably estimate video similarity.

2.2 Similarity search and Dimension Reduction

Given a query video signature, the similarity search problem is to find all the signatures that are close to the query from a very large database. The Database community has long developed sophisticated indexing methods for low-dimensional data, collectively called the Spatial Access Methods (SAM), which can efficiently support similarity search[18, 19]. However, when the data dimension is high, most of these methods become exhaustive search[4]. This problem is commonly known as the "curse of dimension". A common strategy to mitigate this problem is to first reduce the dimension by projecting the high-dimensional data onto a low dimension space where SAM can be applied. Similarity search is then performed on this low-dimensional space to identify potentially similar data points to the query. If the distortion in similarity measurement induced by the dimension reduction scheme is small, such step will eliminate most of the non-similar data points from the search. For the remaining data points, exhaustive search based on the high-dimensional distance is carried out to find the truly similar ones. The most commonly used dimension reduction schemes are the class of linear projections based on Singular Value Decomposition (SVD). SVD is optimal when the distance is Euclidean. For general metric spaces, non-linear methods need to be used. In [20], Faloutsos and Lin propose a heuristic scheme called Fastmap which emulates SVD for general metric spaces. Fastmap is computationally efficient but is later shown to produce non-contractive projections[21], an undesirable property which may lead to early elimination of truly similar data points. The Sparsemap algorithm, proposed by Hristescu and Farah-Colton[22] for indexing protein sequences, only generates contractive projections. The algorithm is based on the celebrated Lipschitz embedding results by Bourgain[23], which provides tight bounds for both the dimension and distortion for the dimension reduction problem in general metric spaces. However, Sparsemap has two two drawbacks. First, the process of computing the low-dimensional projections for each data point is computationally complex. In addition, if the desired dimension is lower than the bound imposed by Bourgain, it is hard to select the optimal projection because the number of possible projections is exponential. In [24], Berman and Shapiro propose a similar scheme called Triangle-Inequality Based Pruning (TIBP) which uses a much restricted search space. I combine TIBP with video signatures and design an algorithm to compute the optimal projection for a given target dimension. I will experimentally demonstrate that my proposed scheme outperforms the Fastmap algorithm.

2.3 Clustering

The Information Retrieval (IR) community has long noticed that clustering highly-correlated, or similar data items can improve the efficiency of an automatic IR system[25]. A clustering structure organizes data so that users can quickly access relevant information. Areas such as browsing and navigation[26, 27] and story segmentation of video sequences[28, 29] are just a few examples where clustering algorithms are extensively used. Another benefit of clustering is its potential gain in retrieval performance over simple thresholding or ranking similarity search. When presented with a query, a simple thresholding search retrieves all items in the database within a certain distance threshold away from the query, while ranking search retrieves a fixed number of items closest to the query. In either case, the relationships among individual data items in the database are completely ignored. On the other hand, a system that employs clustering will first group the entire database into clusters of similar items by considering all possible relationships, and return the cluster closest to the query. Even though the performance varies based on the precise definition of the algorithm, by considering the totality of the data, the process of clustering can typically reduce imprecision in the distance measurement and discover hidden similar items. Recently, many researchers apply clustering techniques to improve

retrieval performance for multimedia retrieval systems[30, 14, 31].

There are myriads of different clustering algorithms in the literature. Many of them are summarized in works by Theodoridis and Koutroumbas[32] and Everitt[33]. Clustering video signatures, however, imposes a number of requirements on the clustering algorithm. First, the number of clusters is very large and unknown a prior. This excludes the class of algorithms, such as k-means clustering, that require the number of clusters as part of the input. Second, the clustering algorithm should be able to adapt to local statistics in order to handle video clips with very diverse contents. Third, because video signature is a randomized algorithm, there is a small probability for the signature distance to be large even when the two video clips are very similar. Thus, the clustering algorithm must be robust enough to discard such erroneous measurements. One class of clustering algorithms that can be used for our problem is the agglomerative hierarchical clustering[32]. This class of algorithms seeks a hierarchical clustering structure by successively combining clusters which are close to each other. Single-link and complete-link clusterings are the two most commonly used hierarchical algorithms. They differ from each other on how they measure the distance between two clusters. Single-link defines the distance based on the two closest elements from the two clusters, while complete-link uses the elements farthest apart. One problem of these algorithms is that they are not very adaptive to data statistics as a fixed distance threshold is typically required to terminate the clustering process.

To this end, I find the class of clustering algorithms based on graph-theoretical concepts to be the most suitable for my problem. This class of algorithms treats data items as vertices in a graph and connects potentially similar items with edges. Instead of using distances alone to form clusters, the algorithms consider the connectivity within the graph and identify highly connected regions as similar clusters. Graph-theoretical clustering techniques are successfully applied to image retrieval systems[30], image segmentation[34] and gene expression clustering [35]. I develop a novel graph-theoretical clustering algorithm for video signatures. My algorithm addresses threshold adaptation and signature uncertainties by considering highly connected regions at many different distance thresholds. I will demonstrate that my clustering algorithm produces better retrieval performance than simple thresholding, single-link and complete-link hierarchical clustering.

3 Video Signature

In order to be robust against temporal editing, a video sequence V is modeled as a collection of its individual frames $\{v\}$. The similarity between video sequences is based solely on the similarity between individual frames. Let $V = \{v\}$ and $W = \{w\}$ denote two video sequences. In all my algorithms, each frame will be represented by a high-dimensional feature vector. Thus, I will use the terms "frame" and "feature vector" interchangeably. Assume that there is a visual feature distance function $d(v, w)$ between frames v and w. In all of my experiments, I use a normalized quadrant-based HSV color histogram with l_1-metric, or sum of absolute differences, to measure similarity[36]. Each quadrant color histogram has 178 bins with 18 bins for hue, 3 for saturation, 3 for value, plus 16 pure gray levels. I define video distance $H(V, W)$ as the average distance between the closest matched frames of the two video sequences:

$$H(V,W) \stackrel{\Delta}{=} \frac{1}{|V|+|W|}\left[\sum_{v\in V} d(v, g_W(v)) + \sum_{w\in W} d(g_V(w), w)\right]$$

where $g_X(y) \stackrel{\Delta}{=} \arg\min_{x\in X} d(x, y)$ denotes the frame in video sequence X which is visually closest to frame y. $|V|$ and $|W|$ denote the size of sets V and W respectively. In practice, it is computationally

prohibitive to compute H because its complexity is proportional to the product of $|V|$ and $|W|$. To reduce the complexity, I introduce a particular form of random sampling called *video signature*.

Let $R = \{s_1, s_2, \ldots, s_M\}$ be a set of M randomly selected images which I call *seeds*. I define the *video signature* of V with respect to R as the M-tuple of frames $V_R \triangleq (v_{s_1}, v_{s_2}, \ldots, v_{s_M})$, where $v_{s_i} \triangleq g_V(s_i)$ is the frame in V closest to the seed s_i. If two video sequences V and W are very similar, their signature frames with respect to an arbitrary seed s, i.e. v_s and w_s, are likely to be similar as well. By using only a small number of signature frames, I will show that comparing video signatures can reliably identify similar sequences. The biggest advantage of this scheme is its simplicity: Given a fixed number of signature frames M, the complexity to generate V_R is proportional to the length of the video.

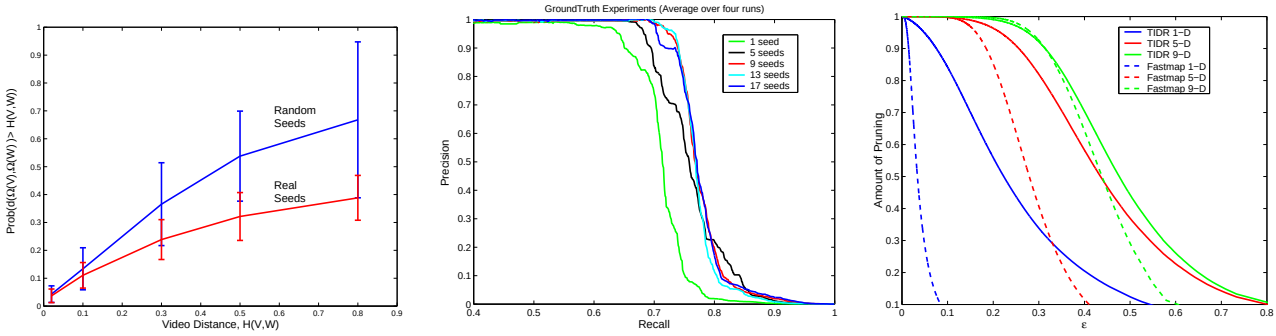
Two types of error may arise when using video signatures to measure similarity. Type I error happens when signature frames are similar to each other while the two underlying video sequences are not. The probability for such an error is typically small. It is rare for two very different video sequences, both from a highly heterogeneous database like the web, to share any similar frames. In addition, I prevent frivolous match by rejecting frames with low information content, such as black frames, to be signature frames. The second type of error, or type II error, happens when two similar video sequences produce non-similar signature frames. This type of error reduce the recall performance as similar sequences cannot be retrieved using their signatures. In [3], I show that the error probability is related to the volume between the Voronoi boundaries of the two similar sequences. This probability typically depends on the geometry of the feature space, the distribution of individual frames, and the distance between the two similar sequences. In general, it is difficult to compute this quantity analytically. Thus, for the color histogram feature used in the experiments, I attempt to characterize this probability by using simulations. I use 15 complex MPEG-7 test sequences with artificial random noise injected to create similar sequences at different distance levels. More details about the simulations can be found in [37].

Figure 1(a) shows the average error probability versus the distance between similar video sequences in the case when a single seed is used. Two types of seeds are compared in my simulations – random seeds and real seeds. Random seeds are generated by uniform sampling on the histogram space. Real seeds represent a set of diverse images from the web. As shown in the figure, the error probability is small when the similar sequences are close to each other, but it gradually increases as the distance increases. The figure also shows that real seeds have much lower average error probabilities and small variances than random seeds. Intuitively, this phenomenon can be explained by the fact that real seeds are much closer to the real video sequences than random seeds. If a seed happens to coincide with one of actual video frames, no error will occur. Thus, seeds closer to the sequences are generally more robust against small local perturbations. For all my subsequent experiments, I use real seeds for signature generation.

The above analysis is performed for a single signature frame only. By using multiple signature frames, it is possible to further reduce the error probability. To balance between two types of error, I aggregate distance measurements between multiple signature frames using the median operator. Given two signatures V_R and W_R, I define the signature distance as follows:

$$D_{sig}(V_R, W_R) \triangleq \text{median}\{d(v_{s_i}, w_{s_i}), i = 1 \ldots M\}.$$

The number of seeds M determines the computational complexity of the signature distance. I determine this quantity based on the retrieval performance of a ground-truth set from the entire web dataset of 46,000 clips. The ground-truth set is determined statistically by combining meta-data, video signatures and expert judgment[6]. Figure 1(b) shows the precision versus recall curves at different



(a) (b) (c)

Figure 1: (a) Average type II error probability at different video distances; (b) Precision versus recall curves for identifying a ground-truth set using different numbers of seeds; (c) Amount of pruning versus distance threshold for optimal TIBP and Fastmap.

numbers of seeds. The precise definitions of recall and precision can be found in [6]. Each point on the curves represents the average of four independent runs. As shown in the figure, both precision and recall improve when more seeds are used to reduce the sampling error. The improvement, however, becomes negligible when more than nine seeds are used. At 90% precision, the nine-seed signature scheme achieves 74% recall. The number of signature frames in a signature will set to nine for all my subsequent experiments.

4 Dimension Reduction

A major step in finding the signature distance, $D_{sig}(V_R, W_R)$, is to compute all the distances, $d(v_{s_i}, w_{s_i})$ between signature frames v_{s_i} and w_{s_i} for $1 \leq i \leq M$. The complexity of the frame distance computation is proportional to the dimension of the feature vector, which is usually quite high in multimedia retrieval applications. To perform a similarity search, it is crucial to reduce the complexity of such computation because it needs to be done for every signature frame in the database. As such, I apply and extend a dimensional reduction technique called the Triangle-Inequality Based Pruning (TIBP) to video signatures for complexity reduction.

The original TIBP, as proposed in [24], uses a fixed "key" set K of feature vectors $\{k_1, k_2, \ldots, k_P\}$, where P is the dimension of the space to which all the feature vectors are projected. The projection of a feature vector v is given by a P-tuple $\Theta_K(v) \stackrel{\Delta}{=} (d(v, k_1), d(v, k_2), \ldots, d(v, k_P))$. The metric defined in this lower-dimension space is the l_∞ distance: $d'(\Theta_K(v), \Theta_K(w)) = max_{1 \leq i \leq P} |d(v, k_i) - d(w, k_i)|$. By the triangle inequality, it is easy to see that $d'(\Theta_K(v), \Theta_K(w)) \leq d(v, w)$, and thus, the projection is contractive. To extend this to the entire signature, I apply the same type of projections to individual signature frames and define the projection of a signature $V_R = (v_{s_1}, v_{s_2}, \ldots, v_{s_M})$ as follows:

$$\Theta(V_R) \stackrel{\Delta}{=} (\Theta_{K_1}(v_{s_1}), \Theta_{K_2}(v_{s_2}), \ldots, \Theta_{K_M}(v_{s_M})).$$

Note that the M key sets $K_1, \ldots, K_M$ can be different from one another. For simplicity, I assume that all the key sets are of the same size P. The distance between projections can be computed as follows:

$$D'_{sig}(\Theta(V_R), \Theta(W_R)) \stackrel{\Delta}{=} \text{median}\{d'(\Theta_{K_i}(v_{s_i}), \Theta_{K_i}(w_{s_i})), i = 1, \ldots, M\}.$$

Despite the fact that neither D_{sig} nor D'_{sig} is a true metric, Θ is contractive because all Θ_{K_i}'s are contractive and the median operator preserves the ordering.

Now, I turn to the problem of selecting the key sets K_i for $i = 1, \ldots, M$. I assume that there exists a large key library $\boldsymbol{K}$ from which I select all K_i's. The goal of any dimensional reduction scheme is to minimize the distortion of the distance measurements caused by the projection. A commonly used cost function to evaluate dimension reduction schemes is the square error function. Assume that $\boldsymbol{V}_i$ is the set of all i-th signature frames. The optimal selection of K_i can be formulated as the following minimization problem:

$$K_i \stackrel{\Delta}{=} \arg\min_{K \subset \boldsymbol{K}, |K|=P} \sum_{v,w \in \boldsymbol{V}_i} (d(v,w) - d'(\Theta_K(v), \Theta_K(w)))^2.$$

It is impractical to directly solve the above minimization problem for two reasons. First, the cost function requires the actual distance values $d(v, w)$ for all pairs of feature vectors in the database. However, computing such distances is precisely the goal of the dimension reduction scheme. If such information is available, there is no need for any dimension reduction. In order to approximate the cost function, I randomly sample a small number of signature pairs to compute their distances. In my experiments, I sample 0.1% of the total number of possible pairs. Second, the number of possible key sets to choose from is $\binom{|\boldsymbol{K}|}{P}$, which is too large to search in practice for any $\boldsymbol{K}$ of reasonable size. As a heuristic, I adopt a greedy approach to incrementally search for the best keys. In each step, I choose the key which produces the maximum reduction in the cost function.

Figure 1(c) shows the performance of my algorithm against Fastmap for $P = 1, 5$ and 9. The key library $\boldsymbol{K}$ consists of 100 color histograms of a diverse set of real images. The goal of this experiment is to identify, from a database of 10,000 signatures, all pairs with signature distance less than ϵ. By using a dimension reduction projection Θ, if the projected distance $D'_{sig}(\Theta(V_R), \Theta(W_R) \geq \epsilon$, the contractive property implies that $D_{sig}(V_R, W_R) \geq \epsilon$[1]. Thus, it is safe to prune away the pair V_R, W_R without actually computing the full signature distance. The graphs in figure 1(c) show the percentages of the pairs that can be pruned away as a function of ϵ for different dimension reduction schemes. As shown in the figure, my algorithm produces significant more pruning than Fastmap for $P = 1$ and 5. For $P = 9$, due to the limited size of the key library, my scheme only provides small improvement over the case for $P = 5$, and performs about the same as the Fastmap with the same dimension.

5 Signature Clustering

In this section, I describe a graph-theoretical clustering algorithm on video signatures. I model the entire database of signatures as a threshold graph $P(\boldsymbol{V}, \mu)$. $\boldsymbol{V}$ is the vertex set denoting all the signatures. There is an edge between any two vertices whose corresponding signature distance is less than μ. I assume μ is large enough such that all similar video sequences have signature distance less than μ. $P(\boldsymbol{V}, \mu)$ can be computed by performing a similarity search for each signature in the database using threshold μ. For a connected component $\boldsymbol{C}$ in the threshold graph, I define the *edge density* $\gamma(\boldsymbol{C})$ as follows:

$$\gamma(\boldsymbol{C}) = \begin{cases} \frac{|E(\boldsymbol{C})| - (|V(\boldsymbol{C})|-1)}{|V(\boldsymbol{C})| \cdot \frac{(|V(\boldsymbol{C})|-1)}{2} - (|V(\boldsymbol{C})|-1)} & \text{if } |V(\boldsymbol{C})| > 2 \\ 1 & \text{otherwise,} \end{cases}$$

[1]Even though Fastmap does not always produce contractive projections, the projected distances computed with Fastmap are all smaller than or equal to the actual distances in my experiments.

$\gamma(\boldsymbol{C})$ is properly crafted such that it evaluates to 0 when $\boldsymbol{C}$ is barely connected, and to 1 when $\boldsymbol{C}$ is complete. I define a *similar cluster* to be a connected component whose edge density exceeds a fixed threshold $\gamma \in (0, 1]$.

The clustering algorithm starts by considering each connected component $\boldsymbol{C}$ of $P(\boldsymbol{V}, \mu)$. If its edge density exceeds γ, the component is declared as a similar cluster and removed from the graph. Otherwise, $\boldsymbol{C}$ is likely to contain a number of distinct similar clusters joined loosely to each other. To recover these clusters, edges are removed in decreasing order of their lengths until some similar clusters emerge. This is reasonable as signatures joined by longer edges are less likely to be similar. This step of edge trimming is equivalent to lowering the distance threshold until the graph is partitioned into multiple connected components. To determine if any of the connected components are similar clusters, their edge densities are computed and compared with γ. The algorithm repeats the process of lowering threshold and checking edge density until all connected components are examined.

The key step of the above algorithm is to find the appropriate distance threshold to partition a connected component $\boldsymbol{C}$ into smaller connected components. A naive approach will be to first sort all the edges based on their lengths, and then remove them in decreasing order until $\boldsymbol{C}$ becomes disconnected. The equivalent distance threshold will be the length of the last edge removed. The drawback of this approach is that connectedness needs to be checked after removing every edge. A simpler method is to make use of a *minimum spanning tree*. A minimum spanning tree or MST $\boldsymbol{T}$ of a connected graph $\boldsymbol{C}$ is a subgraph of $\boldsymbol{C}$ that connects all vertices with the least sum of edge lengths. It can be shown that the longest edge in $\boldsymbol{T}$ will be of the same length as the last edge removed from $\boldsymbol{C}$ before it is partitioned into multiple connected components[38]. Thus, I can first compute the MST $\boldsymbol{T}$ of $\boldsymbol{C}$, and then partition $\boldsymbol{C}$ into connected components by setting the threshold to the longest edge in $\boldsymbol{T}$. The newly-formed connected components are then checked to see if they are similar clusters. Notice that for those components which are not similar clusters, there is no need to recompute MST – the subtrees of $\boldsymbol{T}$ inside these components will be the correct MST. The reason is that if the subtree is not the MST of that component, I can replace that subtree in $\boldsymbol{T}$ with the true MST and obtain a spanning tree for the whole $\boldsymbol{C}$ with lower sum of edge lengths. This contradicts the fact that $\boldsymbol{T}$ is the MST of $\boldsymbol{C}$.

Figure 2(a) shows precision versus recall plots for four algorithms: my clustering algorithm, simple thresholding as described in section 3, single-link, and complete-link hierarchical clustering. The single-link algorithm gives the worst performance. The reason is that as the distance threshold exceeds a certain limit, long chains of non-similar video sequences are erroneously identified as similar clusters. Simple thresholding gives better performance but the single threshold used is inadequate to cater to similar video sequences at different distances. The complete-link algorithm performs quite well at low recall values. Unlike single-link clustering, the complete-link algorithm guarantees that all the video sequences in a cluster are within the distance threshold. Thus, the clustering is more reliable which leads to higher precision values. As the threshold increases to capture similar video sequences that are far apart from each other, the algorithm simultaneously groups some non-similar ones that are relatively close to each other. This situation occurs because a single distance threshold is used in forming clusters. As a result, the precision drops as the threshold becomes too large. My clustering algorithm provides the best performance among all the four schemes and achieves its peak performance of 85% recall and 95% precision with γ equal to 0.3. The precision and recall stay around the same level until γ decreases beyond 0.03. Precision then starts to drop as large loosely-connected regions are identified as clusters. Using $\gamma = 0.3$, I apply the clustering algorithm to the entire web dataset and obtain a total of 6,900 clusters. About 42% of the video sequences in the dataset have at least one similar version. Figure 2(b) shows the distribution of cluster sizes. Some of the largest clusters,

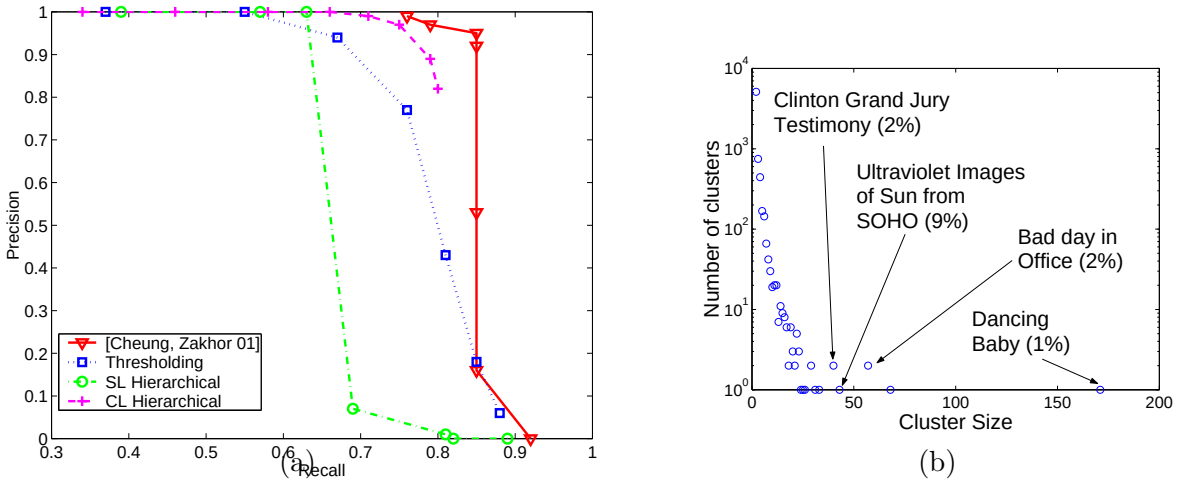


Figure 2: *(a) Precision versus recall for different clustering algorithms and simple thresholding; (b) Distribution of cluster sizes in log-linear scale.*

which represent the highly popular video sequences, are examined and labeled in the figure.

6 Conclusions

The proliferation of video content on the web makes similarity detection an indispensable tool in web data management, searching, and navigation. In my thesis work, I develop a linear-time randomized algorithm to summarize a video sequence into a video signature. By using nine frames per signature, the algorithm achieves 90% precision and 74% recall in identifying a ground-truth set from a large database of web video clips. To facilitate similarity search on video signatures, it is crucial to reduce the complexity in computing high dimensional signature distance. I extend the triangle-inequality based pruning technique to video signatures and apply an optimal criterion in deriving the best dimension reduction scheme. My algorithm substantially outperforms Fastmap in pruning away non-similar pairs in finding similar signatures. To further enhance retrieval performance, I also propose a new signature clustering algorithm. This algorithm outperforms simple thresholding and two hierarchical clustering schemes. At 95% precision, my algorithm attains 85% recall in retrieving the ground-truth set. Applying this clustering algorithm to a dataset of 46,000 web video sequences, the algorithm identifies 6,900 similar clusters, with an average cluster size of 2.81 video sequences.

References

[1] N. Shivakumar and H. Garcia-Molina, "Finding near-replicas of documents on the web," in *World Wide Web and Databases. International Workshop WebDB'98*, Valencia, Spain, Mar. 1998, pp. 204–12.

[2] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, "Trawling the web for emerging cyber-communities," in *Proceedings of the Eight International World Wide Web Conference*, May 1999, pp. 1481–93.

[3] S.-C. Cheung and A. Zakhor, "Estimation of web video multiplicity," in *Proceedings of the SPIE – Internet Imaging*, San Jose, California, Jan. 2000, vol. 3964, pp. 34–6.

[4] R. Weber, H.-J. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," in *Proceedings of the 24th International Conference on Very-Large Databases (VLDB'98)*, New York, NY, USA, Aug. 1998, pp. 194–205.

[5] S.-C. Cheung and A. Zakhor, "Efficient video similarity measurement and search," in *Proceedings of 7th IEEE International Conference on Image Processing*, Vancouver, British Columbia, Sept. 2000, vol. 1, pp. 85–88.

[6] S.-C. Cheung and A. Zakhor, "Video similarity detection with video signature clustering," in *Proceedings of 8th IEEE International Conference on Image Processing*, Thessaloniki, Greece 2001.

[7] B. Perry et al., *Content-based access to multimedia information – from technology trends to state of the art*, chapter 4.3, Kluwer Academic Publishers, Massachusetts, U.S.A., 1999.

[8] D. Adjeroh, I. King, and M.C. Lee, "A distance measure for video sequence similarity matching," in *Proceedings International Workshop on Multi-Media Database Management Systems*, Dayton, OH, USA, Aug. 1998, pp. 72–9.

[9] R. Lienhart, W. Effelsberg, and R. Jain, "VisualGREP: A systematic method to compare and retrieve video sequences," in *Proceedings of storage and retrieval for image and video databases VI*. SPIE, Jan. 1998, vol. 3312, pp. 271–82.

[10] M.A. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding," in *Proceedings 1998 IEEE International Workshop on Content-Based Access of Image and Video Database*, Bombay, India, Jan. 1998, pp. 61–70.

[11] S.-F. Chang, W. Chen, and H. Sundaram, "VideoQ: a fully automated video retrieval system using motion sketches," in *Proceedings Fourth IEEE Workshop on Applications of Computer Vision*, Princeton, New Jersey, Oct. 1998, pp. 270–1.

[12] P. Indyk, G. Iyengar, and N. Shivakumar, "Finding pirated video sequences on the internet," Tech. Rep., Stanford Infolab, Feb. 1999.

[13] M.R. Naphade, R. Wang, and T.S. Huang, "Multimodal pattern matching for audio-visual query and retrieval," in *Proceedings of the Storage and Retrieval for Media Datbases 2001*, San Jose, USA, jan 2001, vol. 4315, pp. 188–195.

[14] H.S. Chang, S. Sull, and S.U. Lee, "Efficient video indexing scheme for content-based retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1269–79, Dec 1999.

[15] B. Günsel, Y. Fu, and A.M. Tekalp, "Hierarchical temporal video segmentation and content characterization," in *Proceedings of the SPIE – Multimedia Storage and Archiving Systems II*, Dallas, USA, 1997, vol. 3229, pp. 46–56.

[16] X. Sun, M.S. Kankanhalli, Y. Zhu, and J. Wu, "Content-based representative frame extraction for digital video," in *IEEE Conference of Multimedia Computing and Systems*, Austin, USA, 1998, pp. 190–3.

[17] A. Girgensohn and J. Boreczky, "Time-constrained keyframe selection technique," *Multimedia Tools and Applications*, vol. 11, pp. 347–358, 2000.

[18] H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, 1989.

[19] C. Faloutsos, *Searching Multimedia Databases by Content*, Kluwer Academic Publishers, 1996.

[20] C. Faloutsos and King-Ip Lin, "Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," in *Proceedings of ACM-SIGMOD*, May 1995, pp. 163–174.

[21] G.R. Hjaltason and H. Samet, "Contractive embedding methods for similarity searching in metric spaces," Tech. Rep. CS-TR-4102, Computer Science Department, University of Maryland, College Park, USA, Jan 2000.

[22] G. Hristescu and M. Farach-Colton, "Cluster-preserving embedding of proteins," Tech. Rep. DIMACS 99-50, Rutgers University, Piscataway, USA, 1999.

[23] J. Bourgain, "On lipschitz embedding of finite metric spaces in hilbert space," *Israel Journal of Mathematics*, vol. 52, pp. 46–52, 1985.

[24] A. P. Berman and L. G. Shapiro, "A flexible image database system for content-based retrieval," *Computer Vision and Image Understanding*, vol. 75, no. 1/2, pp. 175–195, July/August 1999.

[25] K. Sparck Jones and C. van Rijsbergen, "Report on the need for and provision of an "ideal" information retrieval test collection," Tech. Rep. British Library Research and Development Report 5266, Computer Laboratory, University of Cambridge, 1975.

[26] S. Krishnamachari and M. Abdel-Mottaleb, "Image browsing using hierarchical clustering," in *IEEE International Symposium on computer and communications*, July 1999.

[27] A. Vellaikal and C.-C.J. Kuo, "Hierarchical clustering techniques for image database organization and summarization," in *Proceedings of the SPIE – Multimedia Storage and Archiving Systems III*, Boston, USA, Nov 1998, vol. 3527, pp. 68–79.

[28] A. Hanjalic, R.L. Lagendijk, and J. Biemond, "Automated high-level movie segmentation for advanced video-retrieval systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 580–8, June 1999.

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

[30] S. Aksoy and R.M. Haralick, "Graph-theoretic clustering for image grouping and retrieval," in *Proceedings IEEE CVPR.*, June 1999, vol. 1, pp. 63–8.

[31] G. Iyengar and A.B. Lippman, "Distributional clustering for efficient content-based retrieval of images and video," in *Proceedings 1998 International Conference on Image Processing*, Vancouver, B.C., Canada, 2000, vol. III, pp. 81–4.

[32] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 1999.

[33] B.S. Everitt, *Cluster Analysis*, Halsted Press, third edition, 1993.

[34] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug 2000.

[35] R. Shamir and R. Sharan, "Algorithmic approaches to clustering gene expression data," in *Current Topics in Computational Biology*. to be published, 2000.

[36] J.R. Smith, *Integrated Spatial and Feature Image Systems: Retrieval, Analysis and Compression*, Ph.D. thesis, Columbia University, 1997.

[37] S.-C. Cheung and A. Zakhor, "Efficient video similarity measurement and search," *Journal paper in preparation*, 2001.

[38] T. Cormen, C. Leiserson, and R. Riverst, *Introduction to Algorithms*, The MIT Press, Cambridge, Massachusetts, 1992.