A FRAMEWORK FOR MEASURING VIDEO SIMILARITY AND ITS APPLICATION TO VIDEO QUERY BY EXAMPLE

Yap-Peng Tan   Sanjeev R. Kulkarni   Peter J. Ramadge

Department of Electrical Engineering
Princeton University, Princeton, NJ 08544, USA
Email: {yptan, kulkarni, ramadge}@ee.princeton.edu

ABSTRACT

The usefulness of a video database relies on whether the video of interest can be easily located. To allow exploring, browsing, and retrieving videos according to their visual content, efficient techniques for evaluating the visual similarity between different video clips are necessary. We present a framework for measuring video similarity across different resolutions – both spatial and temporal. In particular, the video clips to be compared can be properly aligned through the use of suitable weighting functions and alignment constraints. Dynamic programming techniques are employed to obtain the video similarity measure with a reasonable computational cost. An application to searching MPEG compressed video by example is presented to demonstrate the potential use of the proposed video similarity measure.

1. INTRODUCTION

Just like today's text databases (e.g., the World Wide Web), the usefulness of a video database relies upon whether the video of interest can be found within a reasonable amount of time. As it is not easy to describe video content in words, searching video based solely on text information has its limitations. Furthermore, a text annotation standard for video does not currently exist. Not surprisingly, most existing video contains very limited textual annotation and this annotation is done in a variety of styles. Search techniques which allow exploring, browsing, and retrieving video based on its visual content provide a more intuitive searching approach. They should complement other searching mechanisms, such as keyword search, and make searching large video databases much easier. One of the prerequisites for these content-based search techniques is being able to evaluate the visual similarity between different video clips.

We consider the problem of measuring the visual similarity between two video clips. As video is a medium containing both spatial and temporal information, both of these aspects need to be taken into account in assessing the visual similarity of different video clips.

In the literature on classifying and browsing video clips, much attention has been focused on comparing video shots using representative or key frames [1][2]. Recently, techniques for retrieving video clips by comparing the intensity and motion signatures extracted from the representative frames have been proposed in [3][4]. For example, with a predefined frame comparison strategy, the technique described in [3] can compare video clips which have about the same number of frames or representative frames. Although these techniques have attained certain levels of success in their target applications, we believe that the temporal information in video clips can be further exploited.

To strike a good balance between searching accuracy and computational cost, different resolutions (both spatial and temporal) in the comparison of video similarity may be required for different applications. For example, in searching a unique or distinctive video clip from a video archive, key-frame matching may be enough to achieve the goal; however, in searching a video program for video segments which are similar to a given video clip, a more refined measure of video similarity may be useful. In addition, different resolutions for comparison can be applied to different phases of a video search. The amount of data processed can be substantially reduced by starting with a low-resolution search and then applying finer and finer resolution searches to the candidates found from the previous low-resolution searches.

In this paper we propose a framework for measuring video similarity across different resolutions – both spatial and temporal. In general, two video clips to be compared may not have the same number of frames or representative frames, and similar frames may not be well aligned in time. Nevertheless, the proposed framework can, within certain alignment constraints, identify similar frames from two video clips and obtain a useful video similarity (or distance) measure. Discrete time-warping and dynamic programming techniques are used to determine the video similarity with a reasonable computational cost. An application to the problem of *video query by example* is presented to demonstrate the potential use of the proposed video similarity measure.

2. FRAME AND VIDEO SIMILARITY

We use block-based image color histograms to construct the frame similarity measure. Each video frame (or its down-sampled version, such as a DC image extracted from MPEG compressed video) is divided into B sub-blocks, and the frame similarity is measured by comparing the color histograms of the corresponding sub-blocks. Specifically, the similarity between two video frames f and g depends on the absolute difference of the corresponding block-based image color histograms, defined by

$$d(f,g) \triangleq \frac{1}{2P}\sum_{b=0}^{B-1}\sum_{l=0}^{L-1}|H_f(b,l) - H_g(b,l)| \qquad (1)$$

†The work was completed when the first author was a graduate student at Princeton University.

‡This research was supported in part by an IBM graduate fellowship, the New Jersey Center for Multimedia Research, and the National Science Foundation under NYI Grant IRI-9457645.

where B is the total number of image sub-blocks, L is the total number of color histogram levels, P is the total number of image pixels (we assume the two frames have been normalized to the same number of pixels), and $H_f(b,l)$ and $H_g(b,l)$ are the respective color histogram values for images f and g at sub-block b and histogram level l. The similarity between two video frames can be measured at various resolutions by changing the number of image sub-blocks and color histogram levels. The frame distance measure $d(\cdot,\cdot)$ takes values from 0 to 1, with values closer to 0 indicating more similar frames.

Now we address how the temporal similarity of two video clips can be measured properly. As far as the visual content is concerned, two video clips can be considered similar even if they have different lengths (i.e., total number of frames). For example, the fact that a video clip is played at two different rates does not usually hide the similarity of the content. However, in such cases, simply comparing the visual similarity on corresponding but mis-aligned frames may not yield a useful similarity measure. To assess the similarity between two video clips, we need to properly align the frames to be compared and measure the similarity between each potential pair of aligned frames. For this reason, we consider two video clips $\mathcal{R} = \{r_1 r_2 \ldots r_{N_\mathcal{R}}\}$ and $\mathcal{S} = \{s_1 s_2 \ldots s_{N_\mathcal{S}}\}$ to be similar if there exist alignment functions $u(\cdot)$ $(1 \leq u(n) \leq N_\mathcal{R})$ and $v(\cdot)$ $(1 \leq v(n) \leq N_\mathcal{S})$ such that frame $r_{u(n)}$ is similar to frame $s_{v(n)}$, for $1 \leq n \leq N_\mathcal{T}$, where $N_\mathcal{T}$ is the total number of frame pairs to be compared. Frame $r_{u(n)}$ is considered similar to frame $s_{v(n)}$ if their frame distance measure $d(r_{u(n)}, s_{v(n)})$ is sufficiently small.

We hence propose to measure the distance between two video clips $\mathcal{R}$ and $\mathcal{S}$ by

$$D(\mathcal{R},\mathcal{S}) = \min_{u,v} \sum_{n=1}^{N_\mathcal{T}} w(u(n),v(n)) \cdot d(r_{u(n)}, s_{v(n)}) \qquad (2)$$

where $w(\cdot,\cdot)$ is a weighting function which puts different emphasis on different aligned frame pairs, and $\sum_{n=1}^{N_\mathcal{T}} w(u(n),v(n)) = 1$. Finding the optimal alignment functions $u(\cdot)$ and $v(\cdot)$ that minimize the weighted sum of frame distance measures defined in (2) is an integral part of the process of comparing the two video clips.

Without suitable constraints on the weighting function $w(\cdot,\cdot)$ as well as the alignment functions $u(\cdot)$ and $v(\cdot)$, the computed video similarity measure may not be particularly useful. For example, suppose $w(\cdot,\cdot) = 1/N_\mathcal{T}$, $u(\cdot)$ and $v(\cdot)$ can be any functions, and $d(r_p, s_q) = \min_{m,n} d(r_m, s_n)$. In this case, it is easy to see that the video similarity measure degenerates into $D(\mathcal{R},\mathcal{S}) = d(r_p, s_q)$, i.e., the minimum frame distance of all possible frame pairs, by setting $u(n) = p$ and $v(n) = q$ for $1 \leq n \leq N_\mathcal{T}$. Therefore, a small video distance measure can be obtained from two video clips that are visually very different but happen to have just a single frame in common. To prevent degenerate cases like this, where only a small number of video frames from the two video clips are involved in the comparison, or video frames that are similar in content are repeatedly compared to one another and those that are not similar are skipped, the weighting function and the alignment functions need to be carefully chosen. We shall discuss some useful weighting functions in this section. Constrains on the alignment functions will be addressed in the next section.

One reasonable choice for the weighting function is to relate the weight imposed on the current frame distance measure to the number of frames skipped since the last corresponding frame pair. The greater the number of skipped frames, the more weight should be assigned to the current frame distance measure. Below are some examples of weighting functions which have this property:

1. $w_1(u(n),v(n)) = \frac{\Delta u(n) + \Delta v(n)}{W_1}$
2. $w_2(u(n),v(n)) = \frac{\Delta u(n)}{W_{2u}} + \frac{\Delta v(n)}{W_{2v}}$
3. $w_3(u(n),v(n)) = \sqrt[2]{(\Delta u(n))^2 + (\Delta v(n))^2}/W_3$

with $\Delta u(n) = u(n) - u(n-1)$, $\Delta v(n) = v(n) - v(n-1)$, $u(0) = v(0) = 0$; and where $W_1 = \sum_i (\Delta u(n) + \Delta v(n))$, $W_{2u} = \sum_n \Delta u(n)$, $W_{2v} = \sum_n \Delta v(n)$ and $W_3 = \sum_n \sqrt[2]{(\Delta u(n))^2 + (\Delta v(n))^2}$ are proper normalization factors which make each of the weighting functions have unit sum, i.e., $\sum_n w(u(n),v(n)) = 1$. By choosing proper alignment functions, these weighting functions can lead to a symmetric video distance measure, i.e., $D(\mathcal{R},\mathcal{S}) = D(\mathcal{S},\mathcal{R})$. Note that the video similarity measure so defined is also bounded between the minimum and maximum frame distance measures of all possible frame pairs drawn from the two video clips. This property could facilitate the incorporation of the proposed video similarity measure into a framework for searching images based on an image similarity measure.

3. ALIGNMENT CONSTRAINTS AND DYNAMIC PROGRAMMING

To compute the video distance measure defined in (2), we need to search through a large number of possible frame alignments. Even for two short video clips, this number can be so large that an exhaustive search is computationally intractable. Fortunately, in most potential applications we are interested in comparing video clips subject to certain alignment constraints. For example, when measuring video similarity, the temporal order in which frames are present in each video clip is usually required to be preserved. Together with the weighting function $w(\cdot,\cdot)$, these alignment constraints can measure, in different resolutions, the temporal similarity between the two video clips.

For the alignment functions $u(\cdot)$ and $v(\cdot)$, some useful alignment constraints are given below with $N_\mathcal{T}$ denoting the total number of frame pairs to be compared.

C1: $u(1) = 1$ and $v(1) = 1$

C2: $u(N_\mathcal{T}) = N_\mathcal{R}$ and $v(N_\mathcal{T}) = N_\mathcal{S}$

C3: $u(n) \leq u(n+1)$ and $v(n) \leq v(n+1)$, where $1 \leq n < N_\mathcal{T}$

C4: If $\{u(n), v(n)\} = \{k, l\}$, then $\{u(n+1), v(n+1)\} = \{p, q\}$, where $k \leq p \leq k+z$, $l \leq q \leq l+z$, $|p-k| + |q-l| > 0$, and $z \in \{1, 2, \ldots\}$.

C5: $\alpha \leq (v(N_\mathcal{T}) - v(1))/(u(N_\mathcal{T}) - u(1)) \leq \beta$, where $\alpha \leq 1$ and $\beta \geq 1$.

C6: If $u(k) = u(l)$ (or $v(k) = v(l)$), where $l > k$, then $|v(l) - v(k)| \leq a$ (or $|u(l) - u(k)| \leq b$), where $a \geq 1$ (or $b \geq 1$).

Constraint **C1** requires that the first frames from the two video clips be compared to each other, while constraint **C2** requires that the last frames of the two video clips be compared. Constraint **C3** ensures that the video frames are compared in a proper temporal order. Constraint **C4** specifies the possible frame pairs which can be compared to each other after the current corresponding frame pair. Constraint **C5** restricts the lengths of the two video clips not to be different by more than a certain ratio. Constraint **C6** prevents the situation in which a single frame in one clip is used for comparison with frames which are too widely separated in time in the other video clip. Notice that some of these constraints have an effect on the global alignment of the two video clips (e.g., constraints **C1**, **C2**, and **C5**), and some have more influence on the local alignments (e.g., constraints **C3**, **C4**, and **C6**). Not all these constraints are independent of one another. For example, constraint **C4** implies constraint **C3**. Depending upon the objective, these constraints can be selectively combined to provide different measures of temporal similarity between the two video clips. For example, in searching a large video database for video segments with content similar to an example video clip, constraints **C1** and **C2** cannot be applied together because the locations of the similar video segments are unknown before the search. Instead of constraint **C2**, constraint **C5** can be used together with other suitable constraints to find similar video segments whose lengths are not different by more than a certain ratio when compared to the length of the example video clip. An application of these constraints to *video query by example* will be illustrated in the next section.

With these alignment constraints, evaluating the video distance measure according to equation (2) is equivalent to finding the path of minimum cost in a lattice as shown in Figure 1, where each node in the lattice is attached with the cost of the frame distance measure $d(r_m, s_n)$ between the two associated frames r_m and s_n. The weighting scalar $w(\cdot,\cdot)$, which is imposed on the cost $d(\cdot,\cdot)$ at each node, is actually a measure of the distance between the current and previous nodes along a potential path. For example, the weighting function $w_1(\cdot,\cdot)$ defined in Section 2 measures the normalized Manhattan distance between any two successive nodes along a potential path. Each path in the lattice delineates how frames from two video clips are compared with one another. The alignment constraints restrict the possible ways of going from one node to the next. For example, the paths (a) and (b) shown in Figure 1 satisfy most of the alignment constraints listed above. However, path (c) does not meet the constraints **C2** and **C3**.

With this framework, the optimal path (i.e., the path with minimum cost) can be efficiently found using forward dynamic programming techniques [5]. As an example, we show here how to find the optimal path subject to constraints **C1**, **C2** and **C4** (with $z = 1$), and weighting function $w_1(\cdot,\cdot)$. Let $\mathcal{D}(m,n)$ be the minimum cost of all the legitimate paths that start from node (r_1, s_1) (due to constraint **C1**) and end up in node (r_m, s_n) (i.e., images r_m and s_n are compared with each other). With these choices and due to constraint **C2**, we have $D(\mathcal{R},\mathcal{S}) = \mathcal{D}(N_\mathcal{R}, N_\mathcal{S})$. The minimum cost $\mathcal{D}(m,n)$ for all $1 \leq m \leq N_\mathcal{R}$ and $1 \leq n \leq N_\mathcal{S}$ can be computed according to the recurrence equation,

$$\mathcal{D}(m,n) = \min \begin{cases} \mathcal{D}(m-1,n) & + \quad d(r_m,s_n) \cdot \frac{1}{N_\mathcal{R}+N_\mathcal{S}} \\ \mathcal{D}(m-1,n-1) & + \quad d(r_m,s_n) \cdot \frac{2}{N_\mathcal{R}+N_\mathcal{S}} \\ \mathcal{D}(m,n-1) & + \quad d(r_m,s_n) \cdot \frac{1}{N_\mathcal{R}+N_\mathcal{S}} \end{cases}$$

where $\frac{1}{N_\mathcal{R}+N_\mathcal{S}}$ and $\frac{2}{N_\mathcal{R}+N_\mathcal{S}}$ are the weighting scalars resulting from the weighting function $w_1(\cdot,\cdot)$, $\mathcal{D}(m,n) = \infty$ for $m = 0$ or $n = 0$, and the recurrence equation is initialized by $\mathcal{D}(1,1) = d(r_1,s_1) \cdot 2/(N_\mathcal{R}+N_\mathcal{S})$. After the frame distance measures for all possible frame pairs have been computed, the computation of each $\mathcal{D}(m,n)$ needs only three algebraic operations (each algebraic operation consists of one addition and one multiplication) and two numerical comparisons. Therefore, the optimal path, and hence the video similarity measure between the video clips $\mathcal{R}$ and $\mathcal{S}$, can be obtained with $O(N_\mathcal{R} N_\mathcal{S})$ algebraic operations and $O(N_\mathcal{R} N_\mathcal{S})$ numerical comparisons.

4. AN APPLICATION TO VIDEO QUERY BY EXAMPLE

In this section we show that the proposed video similarity measure can be used to search for video segments directly from a MPEG compressed video that are visually similar to a video clip of interest. We shall denote the query video clip by $\mathcal{Q} = \{q_n : 1 \leq n \leq N_\mathcal{Q}\}$, and the video to be searched by $\mathcal{S} = \{s_n : 1 \leq n \leq N_\mathcal{S}\}$. In general, we have $N_\mathcal{Q} << N_\mathcal{S}$. Since the locations of similar video segments are unknown before the search, each frame in $\mathcal{S}$ can be considered as the starting frame of a video segment which is potentially similar to $\mathcal{Q}$. The alignment constraints **C1**, **C4** and **C5** given in Section 3 and forward dynamic programming techniques are used to locate and align each potential video segment, and hence to obtain the video distance measure. We denote the video distance measure computed from each potentially similar video segment with starting frame s_n as $D(\mathcal{Q}, \mathcal{S}_n)$.

As neighboring video frames are highly similar in content, the computed video distance measure $D(\mathcal{Q}, \mathcal{S}_n)$, as a function of frame number n, generally does not vary too wildly. Figure 2 shows a typical example of the video distance measure $D(\mathcal{Q}, \mathcal{S}_n)$ computed from a 3-minute video sequence with a 100-frame query video clip. In the figure, the minimum point of each valley of the function actually indicates the first frame of one of the five video segments similar to the query video clip $\mathcal{Q}$. Notice that the deeper the valley, the smaller the video distance measure, and hence the more similar the corresponding video segment is to the query video clip.

As the function $D(\mathcal{Q}, \mathcal{S}_n)$ is fairly smooth, its local minima can usually be found without exhaustively sampling values of the function. This can reduce the number of computations required. On the other hand, the distance between two successive samples of function $D(\mathcal{Q}, \mathcal{S}_n)$ needs to be reasonably small so that no similar video clip will be missed. To provide a reasonable tradeoff between the search accuracy and the computational cost, we have designed a progressive searching scheme which can normally locate all the local minima of function $D(\mathcal{Q}, \mathcal{S}_n)$ by first evaluating a set of uniformly-spaced samples of function $D(\mathcal{Q}, \mathcal{S}_n)$, and then performing a finer sampling search between each pair of equal-distant samples where a local minimum is likely to occur. Compared to an exhaustive search, i.e., evaluating every data point of the function $D(\mathcal{Q}, \mathcal{S}_n)$, the progressive search can reduce the total number of the required computations considerably. Our experience shows that most of the similar video clips can be located when the number of frames between each pair of the initial uniformly-spaced samples is not much larger than $N_\mathcal{Q}$, the length of the query video clip. In this case, the number of computations can be reduced by about a factor of $N_\mathcal{Q}$ compared to an exhaustive search.

Figure 4 shows two examples of the results from a video query by example. The experiments were performed on MPEG compressed video and the DC image of each frame was extracted for evaluating the image distance measure without fully decompressing the MPEG video. In each example, the first row contains sample frames of the query video clip, and the other rows contain the aligned frames from sample similar video segments found by the query engine. In the first example, a 100-frame left-to-right fast-break was used to search a 3-minute basketball video. All of the four similar fastbreaks in the video were retrieved successfully. Also, since temporal information was well exploited, none of the three right-to-left fastbreaks contained in the video were identified as similar to the query video clip. In the second example, a 100-frame diving video clip was used to search for similar video segments from a 20-minute video of a diving contest. Seventeen similar video segments of dives performed by different divers were successfully identified. Furthermore, since the proposed video similarity measure can compare video clips with different lengths, the lengths of these retrieved video segments have lengths ranging from 51 to 169. Figure 3 shows the video distance measure $D(Q, S_n)$ computed from the diving contest video with the query diving video clip. The seventeen local minima corresponding to the similar video segments identified can be clearly seen in the figure. In this example, the most similar video segment found is the original query video clip itself and the second most similar one is a slow-motion replay of the query video clip in the same program.

5. CONCLUDING REMARKS

We have presented a framework for measuring the visual similarity, both spatially and temporally, between two video clips. With our algorithm, the two video clips to be compared are properly aligned according to the visual similarity of their individual frames through the use of proper weighting function and alignment constraints. Dynamic programming techniques are employed to obtain the video similarity measure with a reasonable computational cost. An application to searching MPEG compressed video for segments with content similar to an example video clip is presented to demonstrate the potential use of the proposed video similarity measure. The proposed framework can be extended to incorporate other video features, such as other intensity/color/texture features, camera/object motion, and speech/audio information, for assessing video similarity.

6. REFERENCES

[1] F. Arman, R. Depommier, A. Hsu and M-Y. Chiu, "Content-Based Browsing of Video Sequences," *ACM International Conference on Multimedia*, pp. 97-103, Oct 1994.

[2] Hisashi Aoki, Shigeyoshi Shimotsuji, and Osamu Hori, "A Shot Classification Method of Selecting Effective Key-Frames for Video Browsing," *ACM Multimedia*, pp. 1-10, 1996.

[3] Nevenka Dimitrova and Mohamed Abdel-Mottaleb, "Content-Based Video Retrieval by Example Video Clip," *Storage and Retrieval for Still Image and Video Databases V*, Vol. SPIE-3022, pp. 59-70, Feb 1997.

[4] Vikrant Kobla, *et al.*, "Compressed Domain Video Indexing Techniques Using DCT and Motion Vector Information in MPEG Video," *Storage and Retrieval for Still Image and Video Databases V*, Vol. SPIE-3022, pp. 200-211, Feb 1997.

[5] Stuart E. Dreyfus and Averill M. Law, *The Art and Theory of Dynamic Programming*, Academic Press, Inc., 1977.

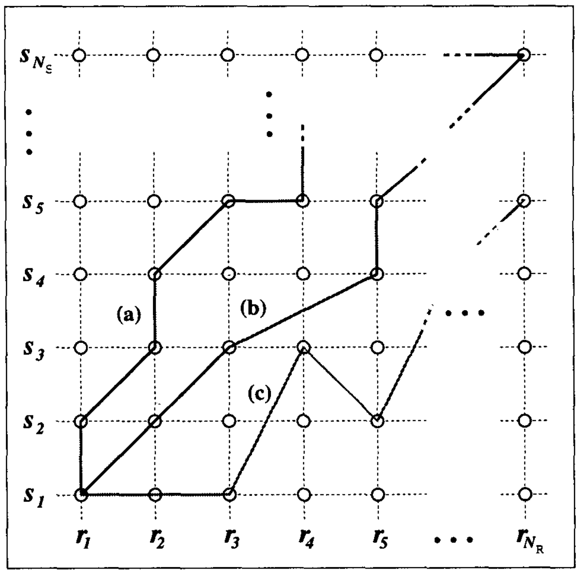

Figure 1: The lattice diagram for comparing two video clips subject to different alignment constraints.

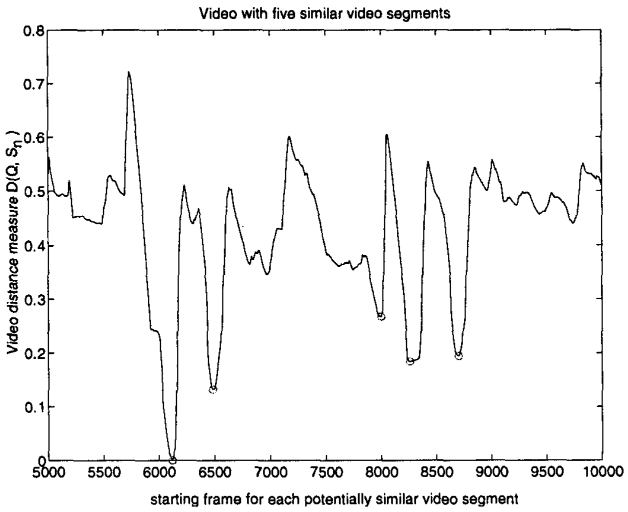

Figure 2: Video distance measure computed from a 3-minute video which has five video segments similar to the query video clip.

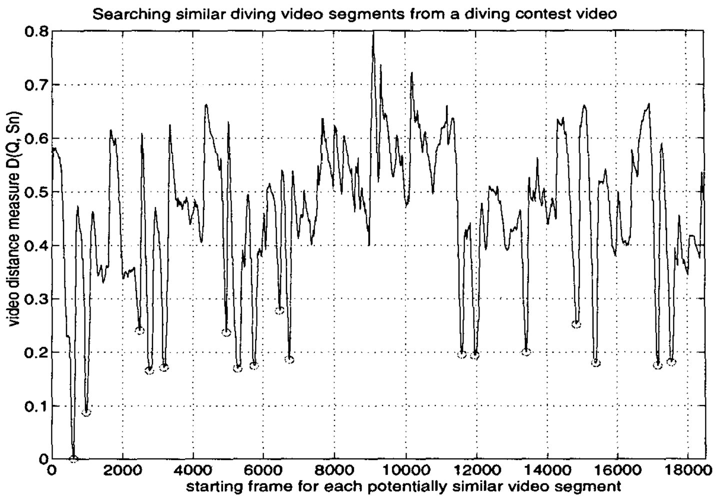

Figure 3: Video distance measure computed from a 20-minute diving contest video with a 100-frame example diving video clip.

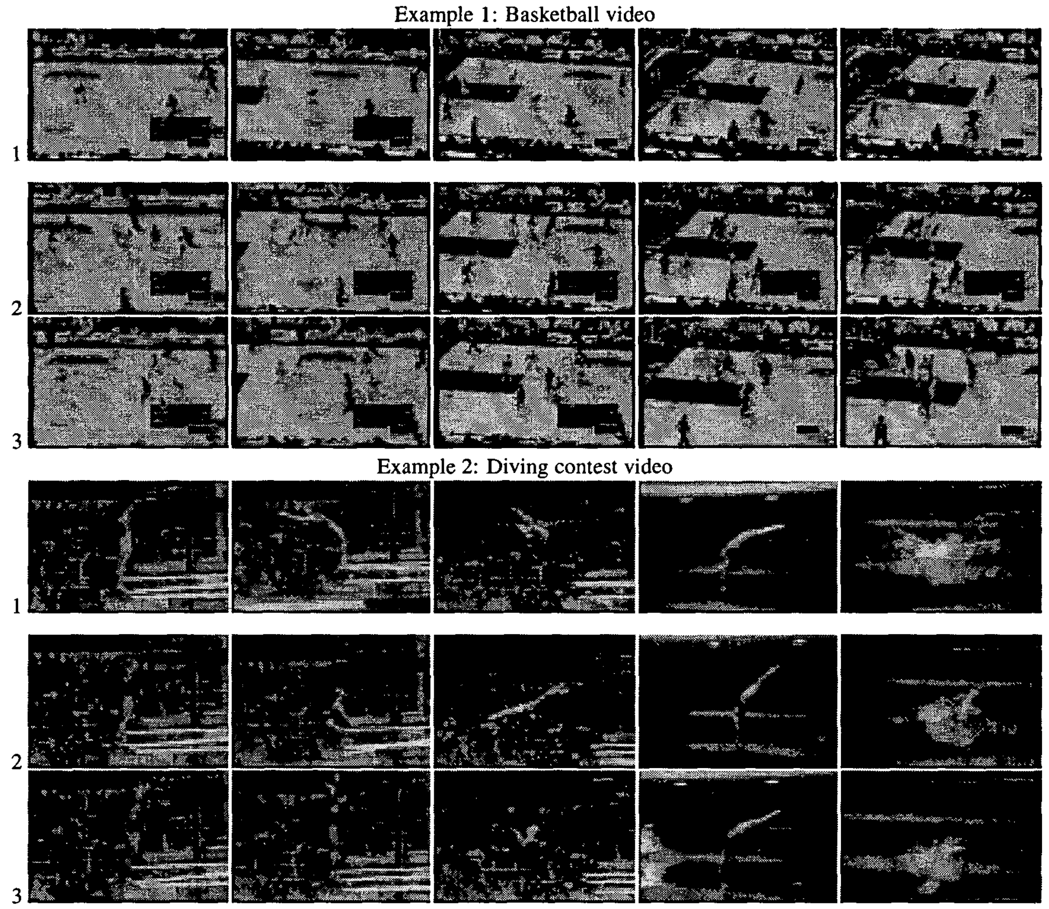

Figure 4: Sample experimental results of video query by example. Clip 1 is the example, and 1,2,3 are the best three rank ordered matches.