EXTRACTION OF KEY FRAMES FROM VIDEOS BY OPTIMAL COLOR COMPOSITION MATCHING AND POLYGON SIMPLIFICATION

Longin Jan Latecki **Daniel de Wildt**
Department of Applied Mathematics
University of Hamburg
Bundesstr. 55, 20146 Hamburg, Germany
latecki@math.uni-hamburg.de

Jianying Hu
Avaya Labs Research
600 Mountain Avenue
Murray Hill, NJ 07974, USA
jianhu@avaya.com

Appeared in *Proc. of Multimedia Signal Processing*, Cannes, October 2001.

Abstract - A video sequence is first mapped to a sequence of points in a semi-metric space that forms a polyline. We require only that a semi-distance between pairs of points be defined that need not satisfy the triangle inequality. By simplifying the polyline, we obtain a small set of the most relevant key frames that is representative of the whole video sequence. The degree of the simplification is either determined automatically or selected by the user. Using our technique, a viewer can browse a video at the level of summarization that suits his patience level. Applications include the creation of a smart fast-forward function for digital VCRs, and the automatic creation of short summaries or trailers that can be used as previews before videos are downloaded from the web.

MOTIVATION

This paper describes an approach to automatically rank video frames by their relevance that depends on the context, i.e., a given frame can be of high relevance in one context but of low relevance in the other, since we would like the rank of the frames to reflect their relevance to the content of the video clip.

For a video sequence, "predictability" is an important concept. If frames are predictable, they are not as important as the ones that are unpredictable. We can rank these frames lower, since the viewer could infer them from context. Frames of a new shot cannot generally be predicted from a previous shot, so they are important. On the other hand, camera translations and pans that do not reveal new objects produce frames that are predictable.

A good candidate to determine the predictability is a distance measure of images. If images n and $n + 1$ in a video sequence have low similarity value, i.e., are very similar, then the image $n + 1$ is predictable with respect to image n.

When applied to the original video sequence, any reasonable image distance measure will give low similarity value for all consecutive frames with except of shot changes. Since there is nearly no significant changes between two consequent frames, the variations in their distance measure are mostly due to noise. What is noise in a video sequence? It is distinct from pixel noise. The image stream generated by a fixed camera looking from a window at a crowd milling around in the street may be

considered to have a stationary component and a visual noise component, due to the changing colors of people's clothes. The passing of a fire truck would be part of the signal over this fluctuating but monotonous background. Due to noise, no threshold value can yield a reasonable set of key frames when applied to the similarity values of images with respect to their neighbor images in an original video sequence.

Therefore, we will use an image distance measure to recursively filter the original video sequence by deleting the most predictable images, until a small set of images (key frames) is left. As the predictability measure of an image B in an image sequence $...ABC...$, we will define $predictibility(B) = |d(A,B) + d(B,C) - d(A,C)|$, where d is an image distance measure. If frames of a video sequence are represented in a vector metric space, $predictibility(B) = 0$ means that frame B lies on line segment AC. The higher is the value of $predictibility(B)$, the further away is B form line segment AC.

In this paper, we first map each image of a video sequence to a point in a semi-metric space, and consequently, the video sequence is mapped to a polygonal trajectory in the semi-metric space. Each image is assigned a compact representation of its color content, i.e., each image is represented as a combination of few most dominant colors that can be viewed as a rough segmentation of the image. This compact representation is used to compute the semi-metric distance between images.

As a result of the first mapping we would like the trajectory to have high curvature for unpredictable scenes and nearly linear parts (due to noise) for predictable scenes. Since we expect the video signal to be noisy in the above sense, we need the second filtering step to enhance the linear parts as well as the parts with a significant curvature. The second filtering step allows a hierarchical output so that the user can specify the level of detail (a scale) at which he wants to view the frames with noteworthy events.

MAPPING AN IMAGE STREAM TO A TRAJECTORY

In this section, we describe the compact representation of color content that is assigned to each image in a video sequence. Then we show how to compute a distance between two images given their compact color representation.

The first step of color distance computation is to obtain compact, perceptually relevant representation of the color content of an image. It has been shown that in the early perception stage human visual system performs identification of dominant colors by eliminating fine details and averaging colors within small areas [8]. Consequently, on the global level, humans perceive images only as a combination of few most prominent colors, even though the color histogram of the observed image might be very "busy". Based on these findings, we perform extraction of perceived colors through the following steps. First a color image is transformed from the RGB space into the perceptually more uniform Lab color space [10]. The set of all possible colors is then reduced to a subset defined by a compact color codebook. Finally a statistical method is applied to identify colors of speckle noise and remap them to the surrounding dominant color (see [5] for details).

Once the perceptually dominant colors are extracted, we represent a color compo-

nent of an image as a pair $CC_i(I_i, P_i)$, where I_i is the index to a color in a particular color codebook and P_i is the area percentage occupied by that color. A color component CC_i is considered to be dominant if P_i exceeds a threshold (typically $2 - 3\%$). Hence, the color composition of an image is represented by the set of dominant color components (DCC) found in the image. Based on human perception, two images are considered similar in terms of color composition if the perceived colors in the two images are similar, and similar colors also occupy similar area percentage [8]. We developed a metric called Optimal Color Composition Distance (OCCD) to capture both criteria [5]. To compute OCCD the set of color components of each image is first quantized into a set of n (typically $20 - 50$) color units, each with the same area percentage p, where $n \times p \approx 100$. We call this set the quantized color component (QCC) set. Suppose we have two images A and B, with QCC sets $\{C_A \mid U_A^1, U_A^2, ...U_A^n\}$ and $\{C_B | U_B^1, U_B^2, ...U_B^n\}$. Let $I(U_x^k)$, $x = A$ or B, $k = 1..n$, denote the color index of unit U_x^k, and $\{M_{AB} \mid m_{AB} : C_A \rightarrow C_B\}$ be the set of one-to-one mapping functions from set C_A to set C_B. Each mapping function defines a mapping distance between the two sets: $MD(C_A, C_B) = \sum_{i=1}^{n} W(I(U_A^i), I(m_{AB}(U_A^i)))$, where $W(i, j)$ is the distance between color i and color j in a given color code book. Our goal is to find the optimal mapping function that minimizes the overall mapping distance. The distance $d(A; B)$ between the images A and B is then defined to be this minimal mapping distance.

This optimization problem can be recast as the problem of minimum cost graph matching, for which there exist well-known solutions with $O(n^3)$ complexity [4]. Note that here n is the number of quantized color components, which roughly corresponds to the maximum number of dominant colors a human being can distinguish within *one* image. It is completely independent of and usually much smaller than the color code book size.

TRAJECTORY FILTERING BY POLYGON SIMPLIFICATION

Our first operation described in the last section maps a video sequence to a trajectory that is a polyline. Since the polyline may be noisy, in the sense that it is not linear but only nearly linear for the video stream segments where nothing of interest happens, i.e., the segments are predictable, and the parts of high curvature are difficult to detect locally, it is necessary to apply the second operation, which we describe the second operation. The goal is to simplify the polyline so that its sections become linear when the corresponding video stream segments are predictable. We achieve this by iterated removal of the vertices that represent the most predictable video frames. In the geometric language for the polyline trajectory, these vertices are the most linear ones. Consequently, the remaining vertices of the simplified polyline are frames that are more non-predictable than the deleted ones.

Our approach to simplification of video polylines is based on a novel process of discrete curve evolution presented in [6] and applied in the context of shape similarity of planar objects in [7]. However, here we will use a different relevance measure of vertices. Fig. 1 illustrates the curve simplification produced by the discrete curve

evolution for a planar figure. Notice that the most relevant vertices of the curve and the general shape of the picture are preserved even as most of the vertices have been removed.

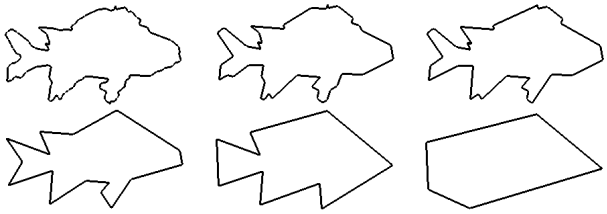

Figure 1: A few stages of our discrete curve evolution.

Let P be a polyline (that does not need to be simple). We will denote the vertices of P by $Vertices(P)$. A *discrete curve evolution* produces a sequence of polylines $P = P^0, ..., P^m$ such that $|Vertices(P^m)| \leq 3$, where $|\ .\ |$ is the cardinality function. Each vertex v in P^i (except the first and the last) is assigned a relevance measure that depends on v and its two neighbor vertices u, w in P^i:

$$K(v, P^i) = K(u, v, w) = |d(u, v) + d(v, w) - d(u, w)| \tag{1}$$

where d is the semi-distance defined in the last section, i.e., it satisfies positivity: $d(x, x) = 0$ and $d(x, y) > 0$ if x is distinct from y, and symmetry $d(x, y) = d(y, x)$, but not the triangle inequality, i.e., there can exist some z's such that $d(x, y) > d(x, z) + d(z, y)$.

The process of *discrete curve evolution* is very simple:

- At every evolution step $i = 0, ..., m - 1$, a polygon P^{i+1} is obtained after the vertices whose relevance measure is minimal have been deleted from P^i.

Observe that relevance measure $K(v, P^i)$ is not a local property with respect to the polygon $P = P^0$, although its computation is local in P^i for every vertex v. This implies that the relevance of a given video frame v is context dependent, where the context is given by the adaptive neighborhood of v, since the neighborhood of v in P^i can be different than its neighborhood in P. Observe also that our relevance measure implies that the length change between P^i and P^{i+1} is minimal if P^{i+1} is obtained from P^i by deleting a single vertex.

EXPERIMENTAL RESULTS

We performed a large number of experimental results to verify the proposed technique using many different kinds of video clips, e.g., commercials, reports from various sport events, and simple synthetic videos. Due to the limited space, we illustrate our results on a single video clip which has a high probability of being seen by the most readers, since knowing the content of the clip is helpful for evaluation of our method.

We present illustrations of the proposed technique for an 80 second MPEG clip from a video named “Mr. Bean’s Christmas”. The clip contains 2379 frames. Fig. 2 shows the obtained storyboard composed of the 10 most relevant key frames corresponding to the vertices of the simplified polyline. In our opinion this summary is very representative for this video clip and contains all relevant frames. These are preliminary results while we are considering comparative benchmarks against ground-truth provided by subjects viewing the clips and selecting small percentages of frames as most descriptive of the stories.

Figure 2: Storyboard with 10 most relevant frames in Mr. Bean’s video (2379 frames).

DISCUSSION AND CONCLUSIONS

In this work, we have proposed and implemented a system for automatically providing short summaries of videos with a frame count that can be controlled by the user or determined automatically. The method is based on a novel fine-to-coarse polyline simplification technique.

Aside from its simplicity the process of the discrete curve evolution differs from the standard methods of polygonal approximation, like least square fitting, by the fact that it can be used in semi-metric, non-linear spaces. The only requirement for discrete curve evolution is that every pair of points is assigned a real-valued distance measure that does not even need to satisfy the triangle inequality.

Observe that even if the polyline representing a video trajectory is contained in Euclidean space, it is not possible to use standard approximation techniques like least-square fitting for its simplification, since the approximating polyline may contain vertices that do not belong to the input polyline. For such vertices, there do not exist any corresponding video frames. Thus, a necessary condition for a simplification of a video polyline is that a sequence of vertices of a simplified polyline is a subsequence of the original one.

In related summarization research, Foote et al. [3] developed browsing tools to help employees access collections of videotaped meetings. Also refer to [9, 11, 12] for other influential work in video browsing research. A lot of efforts is made in the

literature to obtain key frames as different as possible. For example, in [1] after a few complex stages of processing, the last step is to apply an image similarity measure to eliminate similar key frames. In our approach, an image distance that yields an optimal color composition matching of images is elegantly integrated in the filter process. Therefore, we obtain key frames as different as possible for every abstraction level.

References

[1] Y.S. Avrithis, A.D. Doulamis, N.D. Doulamis, and S.D. Kollias. "A Stochastic Framework for Optimal Key Frame Extraction from MPEG Video Databses", *CVIU*, 75:3–24, 1999.

[2] D.F. DeMenthon, V. Kobla, and D. Doermann. "Video Summarization by Curve Simplification", *ACM Multimedia 98*, Bristol, pp. 211-218, 1998.

[3] J. Foote, J. Boreczky, A. Girgensohn, and L. Wilcox, "An Intelligent Media Browser using Automatic Multimodal Analysis", *ACM Multimedia*, Bristol, pp. 375-380, 1998.

[4] H. Gabow. *Implementation of algorithms for maximum matching on nonbipartite graphs*. PhD thesis, Stanford University, 1973.

[5] J. Hu and A. Mojsolovic. "Optimal color composition matching of images," In *Proc. of ICPR*, volume 4, pp. 47–51, Barcelona, 2000.

[6] L. J. Latecki and R. Lakämper. "Convexity rule for shape decomposition based on discrete contour evolution," *CVIU*, 73:441–454, 1999.

[7] L. J. Latecki and R. Lakämper. "Shape Similarity Measure Based on Correspondence of Visual Parts," *IEEE Trans. PAMI*, 22(10):1185-1190, 2000.

[8] A. Mojsolovic, J. Kovacevic, J. Hu, R. J. Safranek, and K. Ganapathy. "Matching and retrieval based on the vocabulary and grammar of color patterns," *IEEE Trans. Image Processing*, 9(1), pp. 38–54, 2000.

[9] M.A. Smith and T. Kanade, "Video Skimming for Quick Browsing Based on Audio and Image Characterization", *Proc. of CVPR*, 1997.

[10] G. Wyszecki and W. S. Stiles. *Color science: concepts and methods, quantitative data and formulae*. John Wiley and Sons, New York, 1982.

[11] M.M. Yeung and B.L. Yeo. "Time-Constrained Clustering for Segmentation of Video into Story Units," *Proc. of ICPR*, 1996.

[12] H.J. Zhang, C.Y. Low, S.W. Smoliar, and J.H. Wu. "Parsing, Retrieval and Browsing: An Integrated and Content–Based Solution," *Proc. of ACM Multimedia*, 1995.