International Symposium on Signal Processing and its Applications (ISSPA), Kuala Lumpur, Malaysia, 13 – 16 August, 2001. Organized by the Dept. of Microelectronics and Computer Engineering, UTM, Malaysia and Signal Processing Research Centre, QUT, Australia

EXTRACTION OF KEY FRAMES FROM VIDEOS BY POLYGON SIMPLIFICATION

Longin Jan Latecki

Department of Applied Mathematics .
University of Hamburg
Bundesstr. 55, 20146 Hamburg, Germany
latecki@math.uni-hamburg.de

Daniel DeMenthon and Azriel Rosenfeld

Center for Automation Research
University of Maryland
College Park, MD 20742, USA
{daniel,ar}@cfar.umd.edu

ABSTRACT

In this paper we apply the polygon simplification method to automatically obtain the most relevant key frames. First a video sequence is mapped to a polyline in $\mathbb{R}^{37}$. By simplifying this polyline, we obtain a summarization (i.e., a small set of the most relevant frames) that is representative of the whole video sequence. The degree of the simplification is either determined automatically or selected by the user.

1. MOTIVATION

This paper describes our 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 that the rank of the frames reflects their relevance to the content of the video clip. At first, it seems that this is hopeless, that we would need to understand the semantic contents of frames and videos. For example, there could be a shot that scans over books on a shelf and stops on the title of a book that is important for understanding the story.

However, in many cases there are syntactic clues. These clues are provided by techniques that the cameraman may use to convey the importance of the moment to the whole story. There may be a zoom, or the camera may stop and "dwell" on an important object, so that the viewer's attention is drawn to the information. In many cases the camera motion corresponds to the motion of the eyes of a surprised viewer. The surprised viewer's gaze is attracted to a strange part of the scene, the gaze scans the scene to "zero in" on it, zooms in on it, dwells on it for a while, until the new information has "sunk in". These changes in the image stream can be detected without understanding of the content.

For an image stream, "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.

We would like to detect when the camera stops (the viewer's gaze stopping on a surprising object). Note that what is unpredictable in this case is the camera motion, not the image content. As the camera slows down, the image content stops changing, so is quite predictable. Therefore, we can consider frames in which the motion field changes as more relevant than the frames where it does not.

Now we present a signal-theoretic view of video summarization. For a video sequence, we have an original signal which is the image stream. We can consider that the image stream has tens of thousands of dimensions if we view each of the three color components of each pixel as a component along a dimension. We apply a first filtering operation. This operation can take the form of a dimension reduction that finds a feature vector for each frame and transforms the image stream into a feature vector trajectory which is a signal in fewer dimensions than the original signal (e.g., 37 in the method described in this paper).

After the first filtering step we would like the trajectory to have high curvature for unpredictable scenes and nearly linear parts (due to noise) for predictable scenes.

What is noise in this context? 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.

Since we expect the video signal to be noisy in this 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 should allow 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.

0-7803-6703-0/01/$10.00©2001 IEEE

2. MAPPING AN IMAGE STREAM TO A TRAJECTORY

We present the first filtering step in this section. As stated in the last section, camera translations and pans should produce feature points that are aligned. Clearly, distances between image frames based on pixel differences are not appropriate since they are sensitive to image translations. On the other hand, distances based on image statistics (histogram, co-occurrence, HMM) are quite insensitive to image translation. To detect unpredictable camera motion, we need to have an image trajectory of high curvature only when such camera motion occurs.

We assign the set of 37 features to each image in a video sequence in the following way:

In the YUV color space that is used in MPEG encoding, for each of the 3 components, we define 4 histogram buckets by dividing the components in 4 intervals. Each bucket contributes 3 feature vector components: the pixel count, and the x and y coordinates of the centroid of the pixels in the bucket. That is 36 components, and we add the time (frame index) to get 37 components. This mapping produces a trajectory that is a polygonal curve in $\mathbb{R}^{37}$.

As the camera translates or pans smoothly without seeing new things, the centroid components change linearly and the trajectory of feature points is linear. If the camera suddenly decelerates, the trajectory has a large curvature, because the centroids decelerate.

An alternative mapping is presented in [2], where a statistical model for each frame is generated using a hidden Markov model (HMM) technique. Then a distance measure between two frames is based on the probability that each frame could have been generated by the model of the other. This distance function defines a *semi-metric space.*

3. TRAJECTORY FILTERING BY POLYGON SIMPLIFICATION

Our first filtering operation (described in Section 2) 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 filtering operation.

In this section we describe the second filtering 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 [7] and applied in the context of shape similarity of planar objects in [8]. 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.

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 $K(v, P^i) \in \mathbb{R}_{\geq 0}$. The relevance measure $K(v, P^i)$ that we used for our experiments is defined below. 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.

Our relevance measure $K(v, P^i)$ that determines the order of vertex deletion depends on vertex v and its two neighbor vertices u, w in P^i. It is given by the formula

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

where d is the Euclidean distance function in R^{37}.

Observe that the relevance measure is not a local property with respect to the polygon P, 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.

We demonstrate with the experimental results in the next section that the discrete curve evolution based on this relevance measure is very suitable as the second filter.

4. 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,

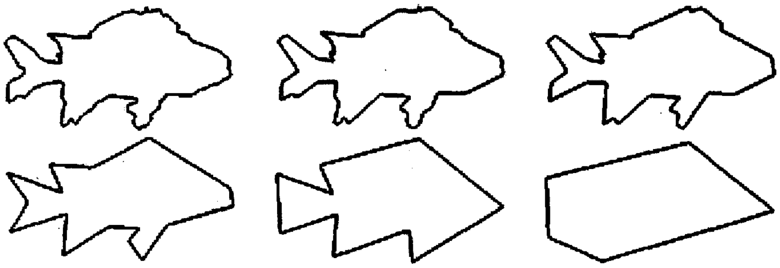
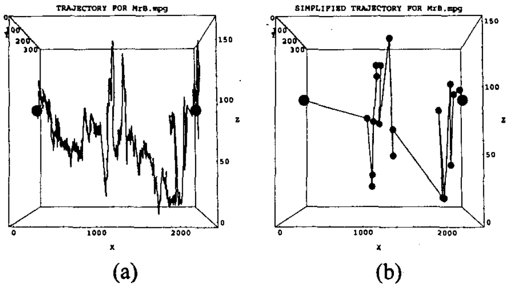


Figure 1: A few stages of our discrete curve evolution.



(a) (b)

Figure 2: (a) Video trajectory with 2379 vertices for Mr. Bean's video clip. (b) A simplified polygon with 20 most relevant frames (black dots).

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 clip from a video named "Mr. Bean's Christmas". The clip contains 2379 frames. First, we map the clip to a video trajectory, which is a polyline with 2379 vertices in $\mathbb{R}^{37}$. A perspective view of the 3D projection of the video trajectory is shown in the left plot (a) of Fig. 2. The two large black dots are the points corresponding to the first and last frame of the video. A curve simplification according to the method of Section 3 was then applied to this trajectory. The middle plot (b) shows a simplified curve in which only 20 points have been preserved. The resulting storyboard composed of the frames corresponding to the vertices of the simplified polyline in plot (b) of Fig. 2 gives a very representative summary for this video clip. We present the 10 most relevant key frames in Fig. 3. In our opinion, summaries obtained by our method are very representative for a large number of video clips to which we applied our method. 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.

5. RELATED WORK AND DISCUSSION

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.

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 and 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. This is, for example, the case if a statistical distance measure is used as similarity measure between images [2].

In [1], we describe how the Ramer method of polygon simplification [10] (called Douglas-Peucker method [3] in cartography circles) could be used to provide summarizations of videos. This method is essentially a recursive binary curve splitting approach that at each recursion splits the arc at the point furthest from the chord, and stops when the arc is close to the chord. This method presents several drawbacks. First, for N video frames it has time complexity N^2, which is prohibitive for large video databases and complex distance measures. Variants have been developed that reduce the complexity to $N \log N$, but they can only be applied to 2D curves, not to multidimensional video trajectories, as they make use of planar convex hulls [6]. Second, the computation of distance between arc and chord requires the use of Euclidean distances. The curve simplification technique we have proposed can be shown to be of order $N \log N$ and can accommodate non-Euclidean distances.

In related summarization research, Foote et al. [4] devel-

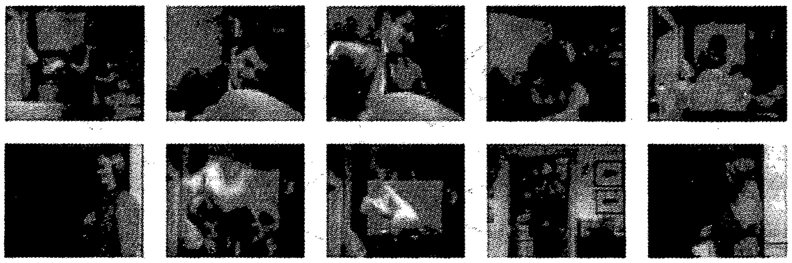


Figure 3: Storyboard with 10 most relevant frames in Mr. Bean's video clip (2379 frames).

oped browsing tools to help employees access collections of videotaped meetings. Also refer to [11, 13, 12, 14] for other influential work in video browsing research.

6. 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 that at each step removes the least relevant vertex and updates the relevances of the affected neighbors. The computation of the relevance measure for each vertex is based on its neighborhood that changes dynamicly during the course of the simplification.

Applications include the creation of a smart fast-forward function to digital VCRs that samples only the most relevant frames, and the automatic creation of short summaries.

7. REFERENCES

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

[2] DeMenthon, D.F., Latecki, L.J., Rosenfeld, A., and Vuilleumier Stückelberg, M., "Relevance Ranking and Smart Fast-Forward of Video Data by Polygon Simplification", Int. Conf. on Visual Information Systems, November 2000, submitted.

[3] Douglas, D.H., and Peucker, T.K.,"Algorithms for the Reduction of the Number of Points Required to Represent a Line or its Caricature", The Canadian Cartographer, 10(2), pp. 112–122, 1973.

[4] Foote, J., Boreczky, J., Girgensohn, A., and Wilcox, L. (1998), "An Intelligent Media Browser using Automatic Multimodal Analysis", ACM Multimedia 98, Bristol, England, pp. 375-380, September 1998.

[5] Jacobs, D., Weinshall, D., and Gdayahu, Y., "Condensing Image Databases when Retrieval is based on Non-Metric Distances", Proc. 6th ICCV, 1998.

[6] Hershberger, J., and Snoeyink, J. "Speeding up the Douglas-Peucker Line-Simplification Algorithm", http://www.cs.ubc.ca/cgi-bin/tr/1992/TR-92-07.

[7] L. J. Latecki and R. Lakämper. Convexity rule for shape decomposition based on discrete contour evolution. *Computer Vision and Image Understanding*, 73:441–454, 1999.

[8] L. J. Latecki and R. Lakämper. Shape Similarity Measure Based on Correspondence of Visual Parts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(10), 2000.

[9] Press, W.H., Teukolsky, S.A., Vettering, W.T., and Flannery, B.P., *Numerical Recipes in C*, Second Edition, Cambridge University Press, 1992.

[10] Ramer, U., "An Iterative Procedure for the Polygonal Approximation of Plane Curves", *Computer Graphics and Image Processing*, 1, pp. 244–256, 1972.

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

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

[13] Yeung, M.M., Yeo, B-L., Wolf, W. and Liu, B.,"Video Browsing using Clustering and Scene Transitions on Compressed Sequences", Proc. SPIE Conf. on Multimedia Computing and Networking, vol. 2417, pp. 399–413, 1995.

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