Motion-based Video Representation for Scene Change Detection

Chong-Wah Ngo, Ting-Chuen Pong*, Hong-Jiang Zhang‡ & Roland T. Chin**

*Department of Computer Science
The Hong Kong University of Science & Technology
{cwngo,tcpong,roland}@cs.ust.hk

‡Microsoft Research China
Beijing, PRC
hjzhang@microsoft.com

Abstract

We present a newly developed scheme for automatically partitioning videos into scenes. A scene is generally referred to as a group of shots taken place in the same site. In this paper, we first propose a motion annotation algorithm based on the analysis of spatio-temporal image volumes. The algorithm characterizes the motions within shots by extracting and analyzing the motion trajectories encoded in the temporal slices of image volumes. A motion-based keyframe computing and selection strategy is thus proposed to compactly represent the content of shots. With these techniques, we further present a scene change detection algorithm by measuring the similarity of the representative keyframes in shots.

1. Introduction

A video usually consists of scenes, and each scene includes one or more shots. A shot is an uninterrupted segment of video frame sequence with static or continuous camera motion, while a scene is a group of shots taken place in the same site. By decomposing a video into scenes, we can facilitate content-based video browsing and summary.

Previous work on scene change detection includes [2, 3, 4]. Basically there are two major approaches: one adopts the time-constraint clustering algorithm to group shots which are visually similar and temporally closed as a scene; the other employs audiovisual characteristics to detect scene breaks. In general, the success of these approaches relies on the video representation scheme and shot similarity measure. The former targets at representing a video in a compact yet semantically meaningful way, while the later attempts to mimic human perception capability. In most systems, shots are represented by a set of selected keyframes, and the similarities among shots are dependent on the color similarity of these keyframes.

In this paper, we propose a motion-based video representation approach with application for scene change detection. The idea is to represent shots adaptively and compactly through motion annotation. For instance, a static sequence is well represented by one frame, a zoom sequence is well described by the frames before and after zoom, while a panning sequence could be summarized by a mosaic image. These computed and selected representative frames are used for shot similarity measure to detect scene breaks.

2. Motion Annotation

Our proposed scheme is based on the motion analysis of spatio-temporal image volumes. We first show the emergence of motion patterns in spatio-temporal slices, and then propose a tensor histogram computation algorithm to describe the motion in a volume. Motion trajectories which are tracked from the histograms, inherently provide clues for temporally segmenting and characterizing both the camera and object motions.

2.1. Temporal Slice Pattern

A video can be arranged as a volume with (x, y) representing image dimensions and t temporal dimension. We can view the volume as formed by a set of $2D$ temporal slices each with dimension (x, t) or (y, t), for example. Each spatio-temporal slice is then a collection of $1D$ scans in the same selected position of every frame over time.

Figure 1 shows various patterns in slices due to camera and object motions. The orientation of the pattern reflects the type of motion. A static sequence exhibits horizontal lines across the horizontal and vertical slices; while camera panning or tilting results in one slice indicating the speed and direction of the motion, and the other slice exploring the panoramic information. For zooming, the lines in slices are either expanded in or out like a V-shape pattern. In addition, a sequence with moving objects generates irregular patterns in different slices. A sequence which tracks an object over time manifests two motion patterns in a horizontal slice, one indicates camera panning and one shows object motion.

0-7695-0750-6/00 $10.00 © 2000 IEEE

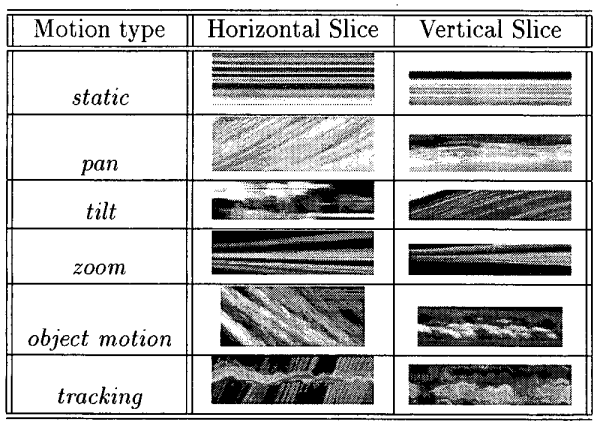


Figure 1: Motion patterns in slices. The horizontal and vertical slices are extracted from the center of an image volume. The x-axis is in time dimension while the y-axis is in image dimension.

2.2. Structure Tensor

The tensor Γ of a slice $\mathbf{H}$ can be expressed as

$$\Gamma = \begin{bmatrix} \mathbf{J}_{xx} & \mathbf{J}_{xt} \\ \mathbf{J}_{xt} & \mathbf{J}_{tt} \end{bmatrix} = \begin{bmatrix} \sum_w \mathbf{H}_x^2 & \sum_w \mathbf{H}_x\mathbf{H}_t \\ \sum_w \mathbf{H}_x\mathbf{H}_t & \sum_w \mathbf{H}_t^2 \end{bmatrix} \quad (1)$$

where $\mathbf{H}_x$ and $\mathbf{H}_t$ are partial derivatives along the spatial and temporal dimensions respectively, w is the window of support which is set to 3×3 throughout the experiments. The rotation angle θ of Γ indicates the direction of gray level change in w. Rotating the principle axes of Γ by θ, we have

$$\mathbf{R}\begin{bmatrix} \mathbf{J}_{xx} & \mathbf{J}_{xt} \\ \mathbf{J}_{xt} & \mathbf{J}_{tt} \end{bmatrix}\mathbf{R}^T = \begin{bmatrix} \lambda_x & 0 \\ 0 & \lambda_t \end{bmatrix} \quad (2)$$

where $\mathbf{R} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$. Since we have three equations with three unknowns, θ can be solved by

$$\theta = \frac{1}{2}\tan^{-1}\frac{2\mathbf{J}_{xt}}{\mathbf{J}_{xx} - \mathbf{J}_{tt}} \quad (3)$$

The local orientation ϕ within a w is computed as

$$\phi = \begin{cases} \theta - \frac{\pi}{2} & \theta > 0 \\ \theta + \frac{\pi}{2} & \text{otherwise} \end{cases} \qquad \phi = [-\tfrac{\pi}{2}, \tfrac{\pi}{2}] \quad (4)$$

It is useful to include a certainty measure to describe how well a ϕ approximates the local orientation of w. The certainty c is estimated as

$$c = \frac{(\mathbf{J}_{xx} - \mathbf{J}_{tt})^2 + 4\mathbf{J}_{xt}^2}{(\mathbf{J}_{xx} + \mathbf{J}_{tt})^2} = \left(\frac{\lambda_x - \lambda_t}{\lambda_x + \lambda_t}\right)^2 \quad (5)$$

and $c = [0, 1]$. For an ideal local orientation, $c = 1$ when either $\lambda_x = 0$ or $\lambda_t = 0$. For an isotropic structure i.e., $\lambda_x = \lambda_t$, $c = 0$.

2.3. Tensor Histogram

A $2D$ tensor histogram $\mathbf{M}(\phi, t)$ is composed of a set of 1D orientation histograms along the time dimension. It is constructed to model the orientation distribution. Mathematically, the histogram can be expressed as

$$\mathbf{M}(\phi, t) = \sum_{\Omega(\phi,t)} c(\Omega) \quad (6)$$

where $\Omega(\phi, t) = \{\mathbf{H}(x, t) | \Gamma(x, t) = \phi\}$, which means that each pixel in slices votes for the bin (ϕ, t) with a certainty value c. The resulting histogram is associated with a confident measure of

$$\mathbf{C} = \frac{1}{T \times M \times N}\sum_{\phi}\sum_{t}\mathbf{M}(\phi, t) \quad (7)$$

where T is the temporal duration and $M \times N$ is the image size. In principle, a histogram with low $\mathbf{C}$ should be rejected for further analysis.

2.4. Motion Characterization

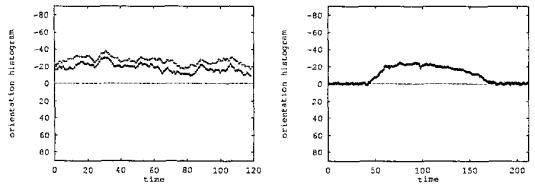
Motion trajectories are traced by tracking the peaks of 1D orientation histograms in $\mathbf{M}(\phi, t)$ over time. A dominant trajectory $p(t) = \max_{-\frac{\pi}{2} < \phi < \frac{\pi}{2}}\{\mathbf{M}(\phi, t)\}$ is defined to have

$$\frac{\sum_{t=k}^{k+15} p(t)}{\sum_{t=k}^{k+15}\sum_{\phi}\mathbf{M}(\phi, t)} > \tau \quad (8)$$

The dominant motion is expected to stay steady approximately for fifteen frames (0.5 second). The threshold value $\tau = 0.6$ is set empirically to tolerate camera jitter. If there is no dominant motion exists, more than one trajectory $p(t)$ will be tracked by a simple path tracing algorithm. The algorithm first looks for $\phi_k = \arg\max_{\phi}\{\mathbf{M}(\phi, k)\}$ at time k, and traces the path for $\phi_{k+1} = \arg\max_{\phi_k - 3 \le \phi \le \phi_k + 3}\{\mathbf{M}(\phi, k+1)\}$. The resulting $p(t)$ should satisfy (8) with $\tau = 0.1$.

These trajectories can correspond to (i) object and/or camera motions; (ii) motion parallax with respect to different depth. Figure 2 shows two examples, in (a) the trajectories correspond to parallax motion, in (b) the trajectory corresponds to static, pan, and static motions over time. Out task is to segment a sequence into smaller units, with each unit being characterized by a motion type.

A sequence with static or slight motion has a trajectory at $\phi = [-\phi_a, \phi_a]$. Ideally, ϕ_a should equal to 0. The horizontal slices of a panning sequence forms a trajectory at $\phi > \phi_a$ or $\phi < -\phi_a$. If $\phi < -\phi_a$, the camera pans to right; if $\phi > \phi_a$, the camera pans to left. A tilting sequence is similar to panning sequence, except that the trajectory is tracked in the tensor histogram



(a) parallax panning (b) static, pan, static

Figure 2: Motion trajectories of tensor histograms.

Motion Type	select or compute
static	one frame
pan or tilt	panoramic image
zoom	first and last image
object tracking	targeted object

Table 1: Motion-driven keyframe selection and computing

generated by vertical slices. Zoom operation, instead of being modeled as a single trajectory, is detected by

$$\frac{\sum_{\phi}\sum_{t>0}\mathbf{M}(\phi,t)}{\sum_{\phi}\sum_{t<0}\mathbf{M}(\phi,t)} \approx 1 \qquad (9)$$

the tensor votes is approximately symmetric at $\phi = 0$. During the implementation, ϕ_a is empirically set to $\frac{\pi}{36}$ (or 5^o degree) to distinguish static motions from pan and tilt motions. After detecting the dominant static, pan and tilt sub-units, (9) is employed to detect zoom operation.

2.5. Video Representation

A motion-driven keyframe computing and selection scheme, as summarized in Table 1, is proposed to represent the content of shots. Figure 3 shows several examples on the computed representative frames. The panoramic images are constructed by warping together the DC images of MPEG videos according to the displacement of the center scans in an image volume. For the object tracking sequence in Figure 3(d), the scans are on the targeted object, as a result, the background is blurred while the object is captured correctly. For multiple motions case as in Figures 3(e)-(g), more than one representative frames are computed.

3. Scene Change Detection

A group of adjacent shots $\{s_m, s_{m+1}, \ldots, s_{n-1}, s_n\}$ is clustered as a scene $\mathbf{S}_k$ if the following conditions are fulfilled

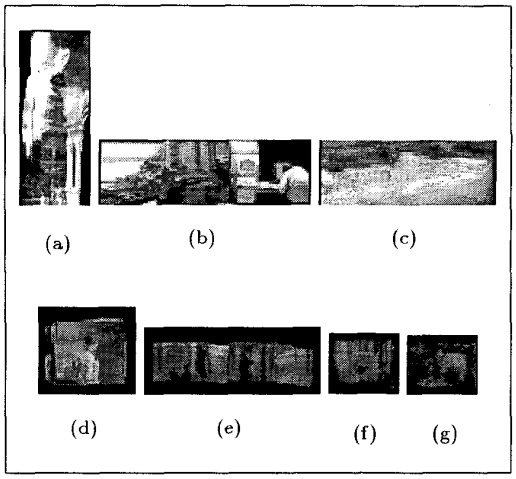


Figure 3: Computed representative frames. (a) panoramic tilt; (b) panoramic pan; (c) parallax pan; (d) object tracking; (e)-(g) multiple motions - panning and tracking.

- *Condition 1*: $\exists t$ such that $t = \arg\{\max_{r=\{1,2,\ldots,c\}} \mathrm{Sim}(s_m, s_{m+r})\}$, $\mathrm{Sim}(s_m, s_{m+t}) \geq T_s$, and $\forall_{r=\{1,2,\ldots,c\}}\ \mathrm{Sim}(s_{m-r}, s_m) < T_s$.
- *Condition 2*: $\exists t$ such that $t = \arg\{\max_{r=\{1,2,\ldots,c\}} \mathrm{Sim}(s_{n-r}, s_n)\}$, $\mathrm{Sim}(s_{n-t}, s_n) \geq T_s$, and $\forall_{r=\{1,2,\ldots,c\}}\ \mathrm{Sim}(s_n, s_{n+r}) < T_s$.
- *Condition 3*: $\exists t_1, t_2$ such that $\{t_1, t_2\} = \arg\{\max_{r=\{0,1,2,\ldots,c\}, s=\{0,1,2,\ldots,c\}} \mathrm{Sim}(s_{i-r}, s_{i+s})\}$, $\mathrm{Sim}(s_{i-t_1}, s_{i+t_2}) \geq T_s$, $m < i < n$ and $0 < |t_1 - t_2| \leq c$.

where $\mathrm{Sim}(s_i, s_j)$ is the similarity measure between the shots i and j and T_s is the similarity threshold. The parameter c is a constraint which is used as follows: suppose $s_j - s_i \leq c$, $i < j$ and $\mathrm{Sim}(s_i, s_j) \geq T_s$, then $\forall_{i \leq k \leq j}\ s_k$ are clustered in one scene.

Condition 1 states that the first shot of a scene must have at least one similar shot succeeding it within the distance c. Similarly, condition 2 states that the last shot of a scene must have at least one similar shot preceding it within c. Condition 3 states that s_i, $m < i < n$, is either similar to a shot preceding or succeeding s_i, or at least one shot preceding s_i is similar to a shot succeeding s_i within c.

Let the representative frames of a shot s_i be $\{r_{i1}, r_{i2}, \ldots, r_{ik}\}$. The similarity between two shots s_i and s_j is defined as

$$\mathrm{Sim}(s_i, s_j) = \max_{p=\{1,2,\ldots\}} \max_{q=\{1,2,\ldots\}} \{\mathrm{Intersect}(r_{ip}, r_{jq})\}$$

Scene	Scene Description	Shots	C	F	M
0	kids learning roller-skater	0-1	1	0	0
1	kids playing in gym	2-13	1	1	0
2	kid playing water	14-24	1	1	0
3	hot balloon even	25-42	1	0	0
4	kids playing on lawn	43-51	1	0	0

Table 2: Experimental results on *lgeraca_lisa_1.mpg*. C: correct detection, F: false detection, M: missed detection.

$\text{Intersect}(r_i, r_j)$, which is the color histogram intersection of the frames r_i and r_j, is expressed as

$$\text{Intersect}(r_i, r_j) = \sum_h \sum_s \sum_v \min\{H_i(h,s,v), H_j(h,s,v)\}$$

where $H_i(h,s,v)$ and $H_j(h,s,v)$ are the normalized HSV color histograms of r_i and r_j respectively. The degree of similarity is proportional to the region of intersection.

The proposed algorithm is similar to [2, 4], except that they did not address the issue of compact video representation for shot similarity measure. Their approaches select image frames as keyframes for similarity measure. Under their scheme, the similarity of two shots is simply computed to be the color similarity of two image frames, which may consequently lead to the occurrence of missed detections.

4. Experiments

We conduct experiments on two MPEG-7 standard test videos: *lgerca_lisa_1.mpg* and *lgerca_lisa_2.mpg*. Both are home videos and each video has approximately 32000 frames. We first employ the video partitioning algorithm proposed in [1] to decompose the videos into shots. Tensor histograms and representative frames are then computed for each shot. These shots are temporally clustered to form scenes by setting $c = 4$ and $T_s = 0.70$. Tables 2-3 show the ground truth data and the experimental results. In *lgeraca_lisa_1.mpg*, the two false alarms are due to illumination effect. In *lgeraca_lisa_2.mpg*, the results of missing five scenes are arguable since these scene are composed of shots in the same places (scenes 4-6 are taken place during gymnastic, scenes 9-11 are taken place on stage, scenes 13-14 are taken place in a swimming pool).

5. Conclusion

We have presented a motion-based video representation scheme for scene change detection. The proposed motion annotation scheme provides compact representation for characterizing shots in videos and measuring similarity in scene change detection. Encouraging results have been obtained through experiments. In future we will develop a more sophisticated scene change detection algorithm based on the identification of background objects.

Scene	Scene Description	Shots	C	F	M
0	kid at home with cat	0-1	1	0	0
1	kids in gym	2-8	1	0	0
2	kids playing high-bar	9-12	1	0	0
3	kids + teacher with high-bar	13-14	1	0	0
4	kids jumping	15-15	1	0	0
5	kids in gym	16-17	0	0	1
6	kids in gym (over-illuminated)	18-28	0	0	1
7	kids playing at home	29-31	1	0	0
8	kid driving outdoor	32-36	1	0	0
9	kids dancing (I)	37-39	1	0	0
10	kids dancing (II)	40-40	0	0	1
11	kids dancing (III)	41-42	0	0	1
12	after play	43-51	1	0	0
13	swimming pool	52-53	1	0	0
14	crowded in swimming pool	54-55	0	0	1

Table 3: Experimental results on *lgeraca_lisa_2.mpg*. C: correct detection, F: false detection, M: missed detection.

Acknowledgments

This work is supported in part by RGC Grants HKUST661/95E, HKUST6072/97E, and HKUST6089/99E.

References

[1] C. W. Ngo, T. C. Pong & R. T. Chin, "Detection of Gradual Transitions through Temporal Slice Analysis," *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 36-41, 1999.

[2] A. Hanjalic *et. al.* , "Automated High-level Movie Segmentation for Advanced Video Retrieval Systems," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, no. 5, pp. 580-88, June, 1999.

[3] J. Huang *et. al.* , "Integration of Audio and Visual Information for Content-based Video Segmentation", *Intl. Conf. on Image Processing.*, vol. 3, pp. 526-9, 1998.

[4] M. M. Yeung & B. L. Yeo, "Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 5, pp. 771-85, Oct, 1997.