

Motion Characterization by Temporal Slices Analysis

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

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

[†]Microsoft Research China
5/F, Beijing Sigma Center
Haidian District, Beijing, PRC
hjzhang@microsoft.com

Abstract

This paper describes an approach to characterize camera and object motions based on the analysis of spatio-temporal image volumes. In the spatio-temporal slices of image volumes, motion is depicted as oriented patterns. We propose a tensor histogram computation algorithm to represent these patterns. The motion trajectories in a histogram are tracked to describe both the camera and object motions. In addition, we exploit the similarity of the temporal slices in a volume to reliably partition a volume into motion tractable units.

1 Introduction

Motion characterization plays a critical role in content-based video indexing. It is an essential step towards creating a compact video representation automatically. We can imagine a camera as a narrative eye, it describes by showing: a camera panning imitates an eye movement to either track an object or to examine a wider view of scene; freeze frames give the impression that an image should be remembered; closeups indicate the intensity of impression. In order to capture these impressions in a compact representation, a panning sequence could be represented by a mosaic image; a static sequence is well represented by one frame; a zoom sequence is well described by the frames before and after zoom, while the focus of a tracking sequence should be the targeted objects. Thus, an effective way of characterizing camera motion in videos will greatly facilitate the video representation, indexing and retrieval tasks.

Related work in this area include camera annotation [2], mosaic representation [3], and motion-layer representations [9, 11]. Bouthemy *et. al.* [2] employed the affine motion parameters to describe dominant camera motions; Irani & Anandan [3] discussed various motion models to annotate and represent videos; while Wang & Edelson [11] and Sawhney & Ayer [9] proposed the motion-based decomposition of videos to describe the background and foreground scenes. Most of these approaches are based on iterative motion parameter es-

timation from two adjacent frames. Generally better results can be acquired if more frames are taken into account at the expense of computational time.

In this paper, we propose an approach based on the spatio-temporal image volume processing [4], which takes into account the larger temporal scale. An image volume is formed by a set of temporal slices which encode rich motion clues suitable for further analysis. Work on the image volume analysis includes the spatio-temporal energy model proposed by Adelson & Bergen [1], video tomography for visualization [10], periodicity analysis [6], and the video partitioning algorithm [7].

In a spatio-temporal image slice, motion is depicted as oriented patterns. Thus, the first part of our work is to compute an orientation map, which we refer to as a tensor histogram, to model the motion distribution existing in the volume. Based on the histogram, an image sequence is temporally segmented into finer units, with each unit consisting of a coherent camera motion. To further model multiple motions within a duration, we exploit the similarity among temporal slices to spatially segment a volume into better units which can describe both camera and object motions. Perhaps the most similar work to our proposed approach is by Joly & Kim [5] who employed Hough transform to detect lines in temporal slices. The orientation of lines reveal the type of motion. Nevertheless, they only select two orthogonal slices for analysis, which in general do not provide sufficient clues for motion annotation. Their work is only applied to the analysis of dominant camera motions. In contrast, our proposed approach analyzes motion clues in the whole image volume and as a result, is capable of temporally and spatially annotating the camera and object motions.

2 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. The slice is used to extract an indicator to capture the motion coherency of the video. For convenience, we refer $\mathbf{H}(x, t)$ as the horizontal slice and $\mathbf{V}(y, t)$ as the vertical slice.

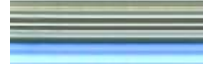











Motion type	Horizontal Slice	Vertical Slice
<i>static</i>		
<i>pan</i>		
<i>tilt</i>		
<i>zoom</i>		
<i>object motion</i>		
<i>tracking</i>		

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.

Figure 1 shows various patterns in slices due to camera and object motions. The orientation of a slice reflects the type of motion. A static sequence exhibits horizontal lines across $\mathbf{H}(x, t)$ and $\mathbf{V}(y, t)$; while camera panning and tilting results in one slice indicating the speed and direction of the motion, and the other slice explores the panoramic information [8]. For zooming, the lines in slices are either expanded in or out in a V-shape pattern. In a multiple motion case, more than one $\mathbf{H}(x, t)$ and one $\mathbf{V}(y, t)$ are, in general, required for analysis. For instance, a sequence with object motion shows both static and panning 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.

3 Motion Analysis

We propose an approach based on the structure tensor computation introduced in [4] to estimate the local orientations of a slice. By investigating the distribution of orientations in all slices, we can classify motion types as well as separate different motion layers.

3.1 Structure Tensor

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

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

where \mathbf{H}_x and \mathbf{H}_t are partial derivatives along the spatial and temporal dimensions respectively. The window of support w is set to 3×3 throughout the experiments. The rotation angle θ of Γ indicates the direction of a gray level change in w . We can rewrite (1) as

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

where

$$\mathbf{R} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

From (2), since we have three equations with three unknowns, θ can be solved and expressed as

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

The local orientation ϕ of a w in slices is computed as

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

It is useful to add in a certainty measure to describe how well 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., $\lambda_x = \lambda_t$, $c = 0$.

3.2 Tensor Histogram

The distribution of local orientations across time inherently reflects the motion trajectories in an image volume. A $2D$ tensor histogram $\mathbf{M}(\phi, t)$ with the dimensions as a $1D$ orientation histogram and time respectively, can be constructed to model the 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

the 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.

Motion trajectories can be traced by tracking the histogram peaks over time. These trajectories can correspond to (i) object and/or camera motions; (ii) motion parallax with respect to different depths. Figure 2 shows two examples, in (a) one trajectory indicates the non-stationary background, and one indicates the moving objects; in (b) the trajectories correspond to parallax motion.

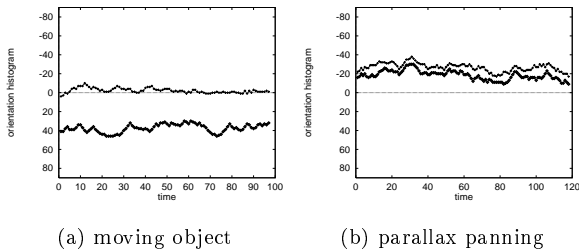


Figure 2: Motion Trajectories in the tensor histograms of image sequences in Figure 1.

The tensor histogram offers useful information for characterizing dominant motions. A sequence with static or slight motion has a trajectory at $\phi = [-\phi_a, \phi_a]$. Ideally, ϕ_a should equal 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 the right; if $\phi > \phi_a$, the camera pans to the left. A tilting sequence is similar to a panning sequence, except that the trajectory is traced in the tensor histogram generated by vertical slices. Throughout the experiments, the parameter ϕ_a is empirically set to $\frac{\pi}{36}$ (or 5° degree). Zoom operation, instead of being modeled as a single trajectory, is detected by

$$\frac{\sum_{\phi} \sum_{t>0} \mathbf{M}(\phi, t)}{\sum_{\phi} \sum_{t<0} \mathbf{M}(\phi, t)} \approx 1 \quad (8)$$

the tensor votes are approximately symmetric at $\phi = 0$.

4 Temporal Motion Segmentation

A video can be partitioned into shots, and a shot can be further divided into finer sub-units, with each

unit indicating a coherent motion. Figure 3 shows the temporal slices of two shots which consist of different motions over time. The corresponding tensor histograms are given in Figure 4. To segment the shots into sub-units, the dominant trajectories are tracked along the temporal dimension. A dominant trajectory $p(t) = \max_{-\frac{\pi}{2} < \phi < \frac{\pi}{2}} \{\mathbf{M}(\phi, t)\}$ is defined to have

$$\frac{\sum_{t=k}^{k+15} p(t)}{\sum_{t=k}^{k+15} \sum_{\phi} \mathbf{M}(\phi, t)} > \tau \quad (9)$$

The dominant motion is expected to stay steady approximately for fifteen frames (0.5 seconds). The threshold value $\tau = 0.6$ is empirically set to tolerate camera jitter. After detecting the static, pan and tilt sequence, (8) is employed to detect the zoom.

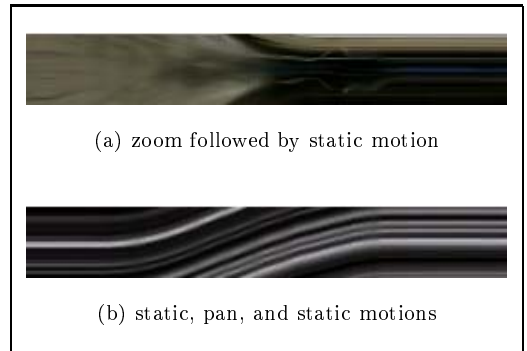


Figure 3: Camera motion changes over time in shots.

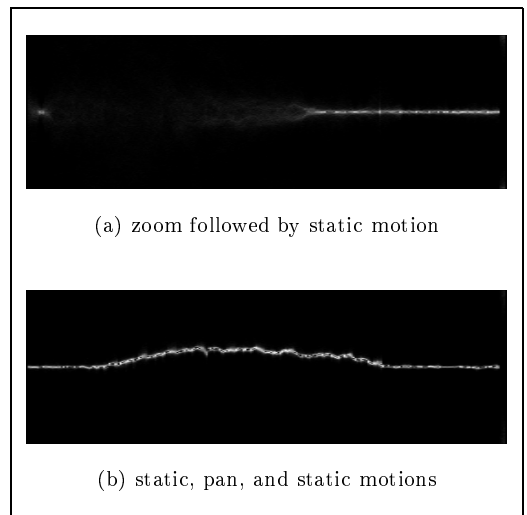


Figure 4: Tensor histograms of the sequences in Figure 3.

4.1 Experiments

To verify the effectiveness of the proposed algorithm, we conduct an experiment on an MPEG-7 standard video, *Nhkvideo.mpg*. The video consists of 15000 frames. We employ the camera break detection algorithm proposed in [7] to partition the video into 45 shots. Our proposed algorithm further divides these shots into finer sub-units according to their motion types. To be computationally efficient, the algorithm operates directly on the DC images of the MPEG video without decompression. Table 1 summarizes the performance of the proposed approach. Throughout the experiment, camera rotation in shot 0 and shot 44 of *Nhkvideo.mpg* is falsely detected as zoom sequences. Similarly, in shot 11 the combination of object rotation and camera tilting is falsely detected as zoom. In shots 7, 20, 31 and 40, the combination of camera pan and zoom results in most of the zooming sequences are falsely detected as pan sequences. In shot 43, the pan sub-unit is undetected since the corresponding slices are mostly occupied by homogeneous regions.

On a Pentium II platform with one processor and 128M main memory, the algorithm takes about 20 minutes to compute the tensor histograms of all shots, and takes less than one second to analyze and classify the camera motion of a tensor histogram. On average, the algorithm processes 12 frames per second. The speed can be further improved by selecting a subset of slices for the tensor histogram computation.

5 Spatial Motion Segmentation

If no dominant motion exists, more than one trajectory $p(t)$ will be tracked by a simple path tracing algorithm. The algorithm first looks for $\phi_k = \arg \max_{\phi} \{\mathbf{M}(\phi, k)\}$ which is the histogram peak at time k , and traces the path for $\phi_{k+1} = \arg \max_{\phi_k - 3 \leq \phi \leq \phi_k + 3} \{\mathbf{M}(\phi, k + 1)\}$. The resulting $p(t)$ should satisfy (9) with $\tau = 0.1$. Two of such examples have been shown in Figure 2.

Intuitively, these trajectories are useful for characterizing multiple motions during a particular time frame. By projecting each trajectory back to the image volume, ideally we can obtain spatially different motion layers. Nevertheless, such projection will generally leave holes in a layer. Filling these holes by extra visual cues such as color and texture is a non-trivial issue. In this section, instead, we propose a more efficient approach by exploiting the similarity information existing in the temporal slices. Based on this information, an image volume is segmented into sub-volumes. For each sub-volume, the tensor histogram is computed to characterize the motion.

To illustrate the idea, we first show an image sequence which involves object tracking in Figure 5. The

shot	static	pan	tilt	zoom
0				F
1		C		
2			C	
3	C			C
4	C			C
5	C	C		
6	C	C		
7		F		M
8	C			
9	C			
10	C			
11				F
12	C			
13	C			
14	C			
15	C			
16		C		C
17	C	C		
18	C			C
19	C	C		
20	C	F		M
21	C			
22	C	C		
23	C			
24	C			M
25	C			
26	C		C	
27	C			
28	C			C
29	C	C		C
30	C	C		
31	C	F		M
32	C			
33	C			
34	C			
35	C			
36	C			
37	C			
38	C			
39				C
40				M
41				C
42				C
43	C	M		C
44				F
45	C			
Recall	1.00	0.90	1.00	0.67
Precision	1.00	0.75	1.00	0.77

Table 1: Motion annotation for the video *Nhkvideo.mpg*. C denotes correct detection; F denotes false detection; M denotes missed detection.

horizontal and vertical slices are shown in Figures 6 and 7 respectively. These slices are extracted from the DC images of size 36×44 . The horizontal slices model the camera and object motions, while the vertical slices explore the background panoramic information as well as follow the target object over time. Intuitively, we want to cluster the horizontal and vertical slices separately so that each cluster represents a motion layer. The clustering criteria is based on the color similarity among slices.



Figure 5: An object tracking image sequence.

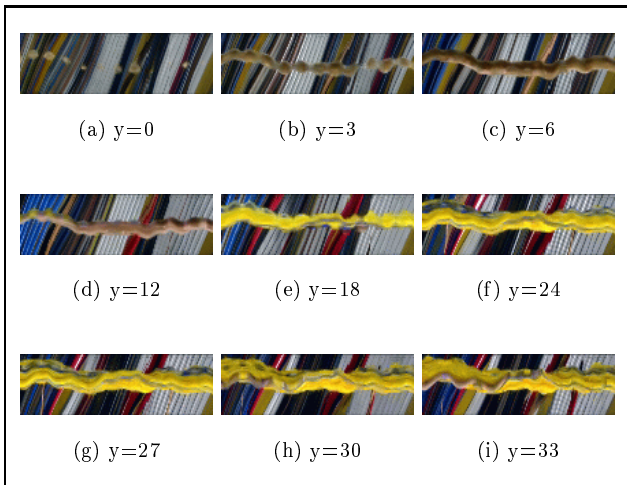


Figure 6: Horizontal slices $\mathbf{H}(x, t)$ of the image sequence in Figure 5.

We employ the color histogram to group similar slices. The hue h is quantized to 18 bins, while the saturation s and brightness v components are quantized to 3 bins respectively. The quantization provides 162 ($18 \times 3 \times 3$) distinct color sets. The similarity between two temporal slices \mathbf{H}_i and \mathbf{H}_j is

$$\sum_h \sum_s \sum_v \min \{D_i(h, s, v), D_j(h, s, v)\} \quad (10)$$

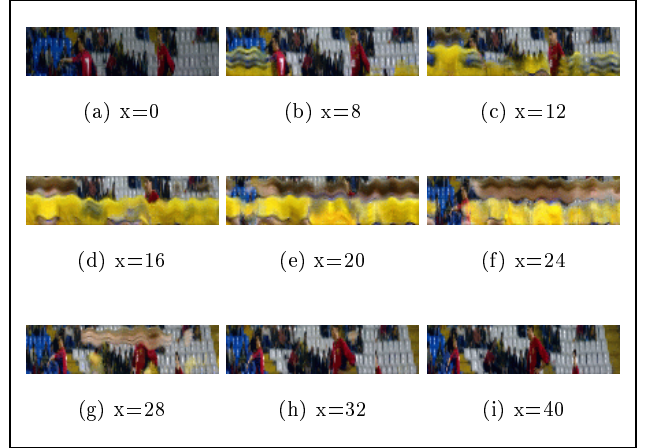


Figure 7: Vertical slices $\mathbf{V}(y, t)$ of the image sequence in Figure 5.

based on the color histogram intersection. $D_i(h, s, v)$ and $D_j(h, s, v)$ are the histograms of \mathbf{H}_i and \mathbf{H}_j respectively. Experimental results show that the horizontal slices are clustered as one group, while the vertical slices are clustered into two groups. By projecting the clustering results into the original image volume, we obtained two sub-volumes. After computing the tensor histograms, one of the sub-volume correctly reflects the camera panning information.

We employ a mosaicking algorithm to illustrate the correctness of the experimental result. The mosaic is constructed by pasting together the DC images based on the displacement computed from the correlation of a few scans in the image sub-volume. Figure 8 shows the mosaicked images; one corresponds with the tracked object, and the other one corresponds to the panning background. The tracked player in Figure 8(a) is blurred due to 3D head and body movements.

We carry out another experiment on a moving objects sequence, as shown in Figures 9. The original image volume is divided into two sub-volumes. The tensor histogram of the moving objects sub-volume resembles a camera panning sequence, as indicated by the temporal slices in Figure 10(b) and (c). The mosaicked image of the moving objects are shown in Figure 11. With the current implementation the total time involved in clustering, tensor histogram computation and mosaicking is approximately 5 frames per second on a Pentium II platform with one processor and 128M main memory.

6 Conclusions

We have presented our work on the motion characterization of videos based on the analysis of spatio-temporal image volumes. On one hand, we propose

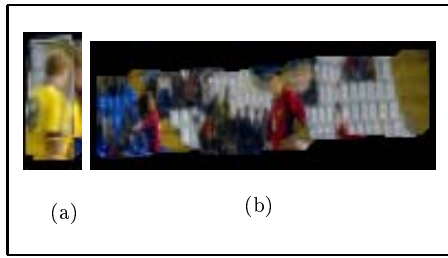


Figure 8: Segmented motion layers of the image sequence in Figure 5, (a) target object; (b) mosaicked background image.

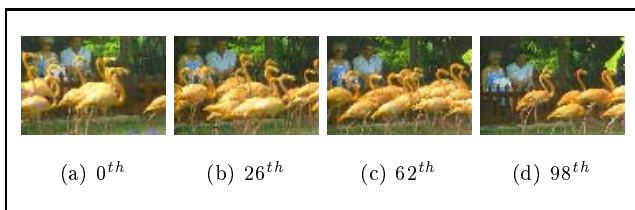


Figure 9: A moving objects sequence.

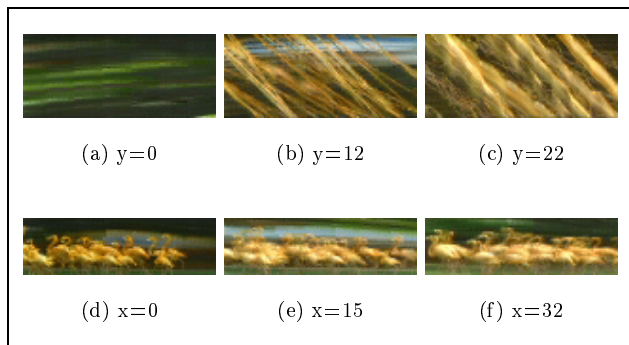


Figure 10: (a)-(c) The horizontal temporal slices; (d)-(e) the vertical temporal slices of the image sequence in Figure 9.



Figure 11: The mosaicked image of the moving objects in Figure 9.

methods to temporally segment a sequence into motion coherent units by tracking its motion trajectories in the tensor histogram. On the other hand, we exploit the similarity of temporal slices to partition the videos into motion layers, with each layer being modeled by a tensor histogram. In the future, we will study the possible ways of applying our work to video browsing and scene change detection. The former application is highly dependent on the video representation techniques, while the later requires more research on background/foreground detection and shot similarity measure.

Acknowledgments

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

References

- [1] E. H. Adelson & J. Bergen, "Spatiotemporal Energy Models for the Perception of Motion," *Journal of Optical Society of America*, vol. 2, no. 2, pp. 284-299, Feb 1985.
- [2] P. Bouthemy *et al.*, "A Unified Approach to Shot Change Detection and Camera Motion Characterization," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, no. 7, pp. 1030-1044, 1999.
- [3] M. Irani & P. Anandan, "Video Indexing Based on Mosaic Representation," *Proc. of the IEEE*, vol. 86, no. 5, pp. 905-921, 1998.
- [4] B. Jähne, *Spatio-temporal Image Processing: Theory and Scientific Applications*, Springer Verlag, 1991.
- [5] P. Joly & H. K. Kim, "Efficient Automatic Analysis of Camera Work and Microsegmentation of Video using Spatiotemporal Images," *Signal Processing: Image Communication*, no.8 pp. 295-307, 1996.
- [6] F. Liu & R.W. Picard, "Finding Periodicity in Space and Time," *Proc. IEEE Intl. Conf. on Computer Vision*, pp. 376-383, 1998.
- [7] C. W. Ngo, T. C. Pong & R. T. Chin, "Detection of Gradual Transitions through Temporal Slice Analysis," *Computer Vision and Pattern Recognition*, vol. 1, pp. 36-41, 1999.
- [8] S. Peleg & J. Herman, "Panoramic Mosaics by Manifold Projection," *Computer Vision and Pattern Recognition*, pp. 338-343, 1997.
- [9] H. S. Sawhney & S. Ayer, "Compact Representations of Videos Through Dominant and Multiple Motion Estimation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 814-830, 1998.
- [10] Y. Tonomura, A. Akutsu, K. Otsuji & T. Sadakata, "VideoMap and Video SpaceIcon: Tools for Anatomizing Video Content," INTERCHI, pp. 131-136, 1993.
- [11] J. Wang & E. Adelson, "Layer Representation for Motion Analysis," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 361-366, 1993.