Vol. 39 No. 5 Transactions of Information Processing Society of Japan May 1998

Regular Paper

VIDEOSTYLER: Video Interface for Spatiotemporal Interactions Based on Multi-dimensional Video Computing

AKIHITO AKUTSU,† YOSHINOBU TONOMURA† and HIROSHI HAMADA†

We propose "VideoStyler", a new video interface made possible by "multi-dimensional video computing". VideoStyler realizes spatiotemporal visualizing and interaction with video content and context according to the user's demands. We introduce the video-processing framework, which is called multi-dimensional video computing, for analyzing a video, creating new structures of the video contents. This paper focuses on two types of video structures that reflect the user's purpose and the kind of video. Video visualization and interaction according to these structures are also discussed. The proposed space oriented visualization and interaction scheme realizes panoramic representation, synthetic representation, and direct access through iconic selection. We implement these functions in a video interface and discuss the efficiency of the proposed interface for grasping and understanding video contents directly and intuitively.

1. Introduction

Because digital video is becoming increasingly important for the networked multimedia society, the audio-visual access environment should allow us to do more than just passively watch. When we try to extract information from a video, we are forced to watch it in real time. We can fast forward or reverse it, but this provides only a very rough overview of the video contents. It is irritating to replay a sequential tape because of the time taken and the many unsuccessful attempts needed to locate a specific segment. Therefore, a method for fast video browsing that enables the viewer to grasp the idea of a lengthy video without watching it in its entirety is one of the most desired functions. In addition to the above-mentioned problems, we are compelled to watch the video as set by the director and/or producer. To scan a video quickly, we need a more direct and comprehensive interface than fast forwarding and normal viewing.

The user interface desired requires video computing which can accurately and rapidly process videos. In video computing studies, the most important issues are 1) how to extract a video's inherent features as unconsciously perceived by people, such as scene changes, camera operations and so on, and 2) how to transform/combine the features to realize new visual representations.

Works on video interfaces with a content-oriented visualization and interaction have been published. Brondmo and Davenport[1] introduced the micon (moving icon), an icon that displays moving images to represent video contents. They used the micon to highlight the video sections of a hypermedia journal of a boat cruise on the Charles River. Tonomura and Abe[2] proposed a content-based visual interface using video icon which is based on a structured icon model using information (cut points, shot length and so on) about video contents. Ueda, et al.[3] proposed an intelligent editing support system called IMPACT that used image analysis to achieve cut detection, camera operation estimation and so on. In this system, they also used micon and visual representations for video interface. Mills, et al.[4] proposed a hierarchical video magnifier that offers a range of views with coarse to fine temporal resolutions. Elliot and Davenport[5] proposed a video interface, called the Video Streamer, which shows spatiotemporal video streams in a box-style volume. Tonomura, et al.[6] proposed a new video iconic representation of spatiotemporal information of video, called VideoSpaceIcon. Teodosio, et al.[7] proposed a new video representation, called Salient Video Stills, which visualized the spatiotemporal information held in a video. Spatiotemporal based video representations offer rich visual cues to the viewer. Arman, et al.[8] proposed content-base video browsing which displays micons and displays motion movement at the edge of frames and the duration of video segments. Kelly, et al.[9] presented an interactive multi

† Nippon Telegraph and Telephone Corporation

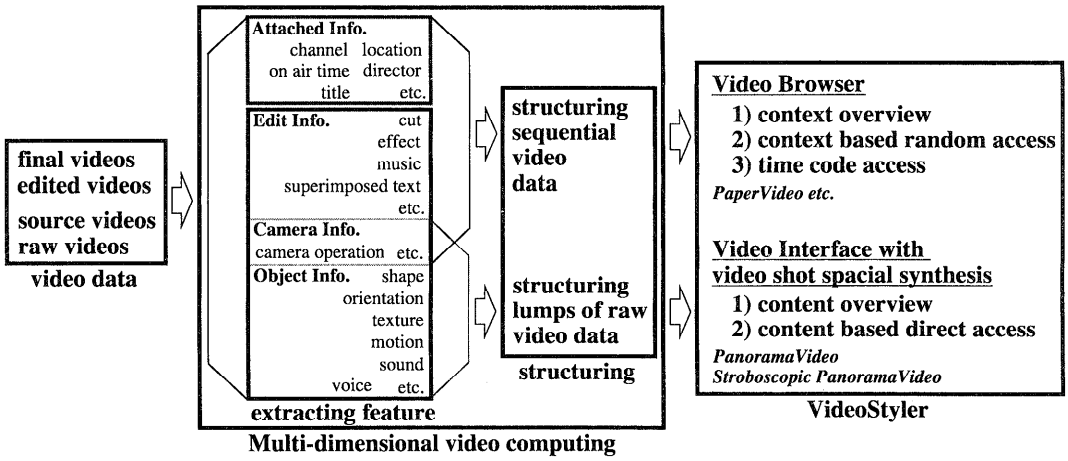

Fig. 1 VideoStyler based on multi-dimensional video computing.

video system, called MultiplePerspective Interactive Video (MPI-Video) which integrated visual computing operations along with 3D modeling and visualization techniques to provide automated analysis and interactive access to data coming from multiple cameras. Kanade, et al.[10] proposed a new visual medium called virtualized reality which delayed the selection of viewing angle till view time; it allows a user to move freely in the scene, independent of the transcription angles used to record the scene.

This paper proposes "VideoStyler", a more direct, efficient and comprehensive video interface with spatiotemporal visualization of and interaction with video contents. It provides not only simple interactive video interfaces, but also a variety of user-selectable video viewing styles. VideoStyler requires "multi-dimensional video computing" processes and so we discuss the issues of video analysis, video structuring, visualization.

First we clearly present the video features extracted by multi-dimensional video computing, and describe the two types of video structures appropriate for developing video interfaces in Section 2. In Section 3, we describe the user interfaces based on the two structures. In Section 4, we propose and discuss the space oriented video interface that offers direct and effective interaction with video content.

2. Multi-dimensional Video Computing

VideoStyler is based on multi-dimensional video computing which offers a video processing framework for analyzing a video, creating new structures, and restyling and visualizing the video according to the user's demands.

Figure 1 shows the basic framework of VideoStyler based on multi-dimensional video computing. Since any final video and/or raw video has some inherent features, such as scene changes, camera operations, color, and sound, the first step is to analyze the inherent features and extract them; this is the multi-dimensional computing process of video parsing. New higher-order links among the features are established to create establish new information structures. Link and content structures allow the user to access the video as desired. The final step is to create appropriate graphic representations for direct and intuitive interaction.

2.1 The Extracted Inherent Features

Figure 2 shows a flow model of the video-making process and examples of inherent features extracted in multi-dimensional video computing. The first stage is gathering source material with a video camcorder. The camcorder captures segments of the real world and each shot packs an infinite amount of information into a finite image plane and audio track. The camera's start and stop operations generate segments of continuous video called "takes". Object information (such as shape, color, motion, voice and so on), and background information (such as scene, sound and so on), and camera conditions (such as start point and end point, panning, tilting, zooming and so on), are important bits of information at this stage. Next, in the editing stage, source video materials are collected, examined and selected. The

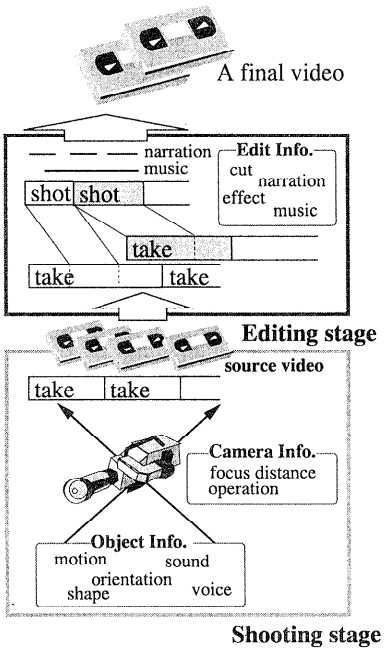

Fig. 2 A flow model of the video-making process.

selected takes are trimmed to "shots", and are linearly edited into a single video stream. Mood music, narration, captions and so on may be superimposed on the concatenated shots. The splices between shots are also called cuts. The final sequence reflects the orders given by the director to the camera operator and the editor. Rich information like object information, cut points, camera operations, and music are embedded in the video. Moreover, the video stream has the implicit structure of the shot sequence. The implicit structure is called "content and context structure".

2.2 Related works for extracting video features

The important issue in multi-dimensional video computing is to extract the essential video features added by production. Automatic video feature extraction algorithms and applications using extracted them have been proposed by many researchers. Research related to multi-dimensional computing have addressed three issues: (1) Temporal segmentation, (2) Motion analysis and transformation and (3) Sound analysis.

(1) Temporal segmentation: Multi-dimensional video computing for scene boundary annotation
To detect scene changes (cut points), several similarity measures between consecutive frames were developed such as the difference between intensity histograms[11] and the difference between color histograms[12]. A twin-comparison method to detect gradual scene changes, including fades, cross-dissolve and wipes was proposed[13]. Another scheme matches video shots and clusters them by considering the temporal variations within individual shots[14].

(2) Motion analysis and transformation: Multi-dimensional video computing for camera operations and scene representative annotation
To extract the spatiotemporal correlation between frames, several approaches have been proposed: optical flow[15], video X-ray[16], affine transformation[17], chirp transformation[18] and 2D and 3D model-based parallax[19].

(3) Sound analysis: Multi-dimensional video computing for auditory scene annotation
A number of studies on the analysis of auditory information have been made considering human perception to sound[20] but only few attempts have so far been made at video analysis using auditory information[21]. Recently, a new method that detects the presence of music and/or voice from a video was proposed[22].

2.3 The Video Structuring For Video Interface

The content and context structures appropriate for video interface construction depend on the user's purpose and the kind of video. Video structuring is classified into two types, 1) structuring sequential video data such as movies, dramas, news, etc. and 2) structuring lumps of raw video data such as multi-camera shots, unedited raw video data and so on.

Sequential video data implicitly has a hierarchical structure that is created by the director during production. The structuring of this type of video data uses edit information that include cut points, music part, narration area and so on. The video browser mentioned in Section 3.1 is appropriate for this type of video data. The video browser is aimed at providing not only effective video browsing, but also a variety of user-selectable video viewing styles to match the user's demands.

The potential structure of a lump of raw video data is formed by the linkage of seg-

mented video clips. Clips are linked according to attributes, for example, several shots which are taken by multi-camera are linked by spatial and/or temporal location, object information, background scene and sound and so on. The interface that uses a content-relation based structure is aimed at intuitively understanding spatial information (for example object locus, size, absolute motion and so on) from the segmented video clips. In Section 3.2, we introduce the VideoStyler interface for handling this type of video structure.

3. User Interface for Video

3.1 Video Browser

The video browser is appropriate for sequential video data with a hierarchical structure. It provides an interface to visualize content and context structure and allows us to grasp video contents intuitively. Automatically extracted cut points are used to visualize the context structure. It is assumed that the cut points are features added by the director and are essential video units. The visualization is realized by showing key frames and/or 3D icons (2D image + 1D time) selected and/or created using cut points from each shot. To display the key frames and/or icons along a time line visualizes not only the content structure, but also the context structure. We have developed several video interfaces with browsing facilities[23),24)]. They are described below. "PaperVideo" is an innovative paper-based video interface in which video information is fixed on paper. The PaperVideo system prints on paper representative images of the video. These representative images are extracted by using the cut point information contained in the video. Because paper is convenient and easy to use, PaperVideo allows us to grasp video contents and context easily.

We have also realized an effective access interface for the large volumes of video data held in digital libraries. The interface can provide different viewing styles to match the user's purpose. The proposed video interface realizes hierarchical interaction where the viewing granularity can range from course to fine. For creating the hierarchical structure, we use attribute information (on air time, production place, etc.) and extracted information (cut points, music, voice points, etc. in video).

Figure 3 visualizes our proposed video browser. We can see a large volumes of video data in overview in the coarsest layer. The

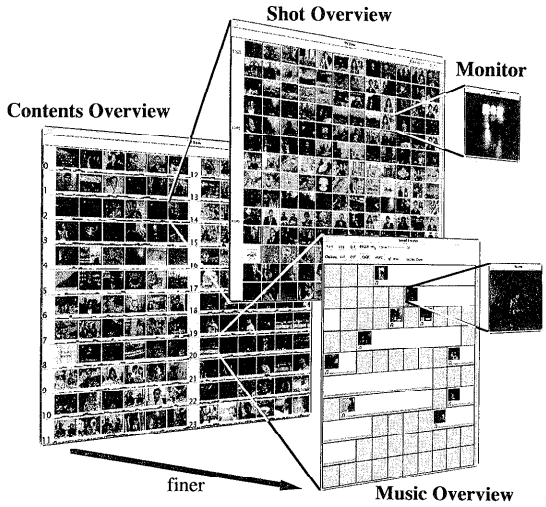

Fig. 3 Video browser for the sequential video data.

coarsest layer is constructed using attribute information. The middle layer uses the extracted information. In the fine layer, we can watch the video contents directly on the monitor.

3.2 Video Interface with Video Shot Spatial Synthesis

Generally speaking, in a video shot where object and camera motion are prominent, one representative image (like a video icon) is not sufficient for grasping the shot contents intuitively. In addition, it is not so easy to understand temporal change in the shot, and the difference of actions in different shots that are related to each other. For example, we can not compare the forms of two ski jumper's intuitively from a normal video of the competition. Usually a numerical record is used for this purpose.

The space oriented video interface supports our grasping and/or understanding of the spatial information in a shot and among segmented video clips intuitively. In this interface, multiple shots are synthesized spatially. The interface allows us to grasp object's locus, size, absolute motion, trajectory, etc. as contained in a shot and in multiple shots.

The following requirements should be satisfied in the space oriented video interface to access spatiotemporal information intuitively.

(1) reconstruction of wide video scene space from a video.
(2) visualization of object motion and trajectory in scene space.
(3) random access to spatiotemporal information by using a spatial index (visual cue).

For (1), a wide video space, like a panoramic space, must be reconstructed from the video. For creating an even wider video space, we spatially synthesize wide video spaces. For (2), the extracted object is fixed in a wide reconstructed space for visualizing object motion and trajectory. The above visualizations are used as an index for realizing direct random access to spatiotemporal information. The video interface that offers (1), (2) and (3) allows us to grasp spatiotemporal video information spatially, and to access the information directly.

In this paper, the video scene space is generated from automatically extracted camera operations which include the information of the spatial relationship between frames in a shot[25]. We create a stroboscopic panoramic image in which an automatically extracted moving object is fixed[26]. The space-based video interface with these panoramic representations permits random access to spatiotemporal information.

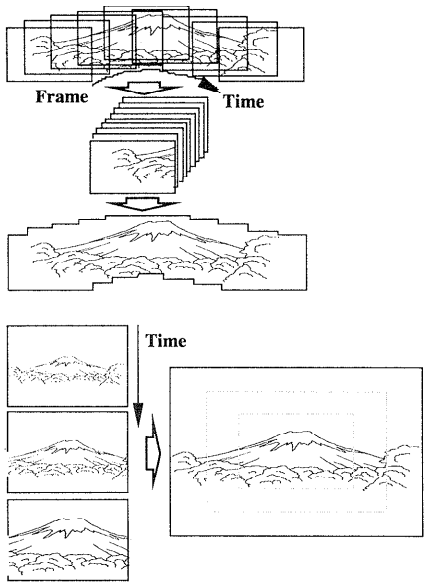

Fig. 4 How to make a panoramic scene space.

4. Implementation of Space-based Video Interface

We collected several video sequences captured by a video camcorder mounted on a tripod. The video sequences held information of a common space, but the temporal information differed because the videos were captured at different times. We realized a space oriented video interface by video spatial synthesis.

4.1 Space Oriented Visualization

4.1.1 Space Reconstruction

The panoramic scene space is created by changing the image position and size of each frame according to the camera operations (**Fig. 4**). We extracted the camera operations information by the VideoTomography method[16]. We call this spatial synthesized scene space PanoramaVideo. A normal panoramic photography has only 2 dimensions, but the PanoramaVideo offers 3 dimensional data (2D image + 1D time).

Figure 5 shows the Panorama Video produced from several shots captured with pan and tilt camera operations. This visualization of shot contents allows us to understand spatial information intuitively without replaying the video repeatedly. Next, we created a spatial synthesized scene space which was wider than PanoramaVideo by using the information of camera operations and the spatial relationship among shots. The affine transform model can be used to approximate the relationship between reconstructed panoramic scene spaces, because the scene spaces on the PanoramaVideo which are made to synthesize were captured by same camcorder on a fixed tripod. The affine transform model is expressed as,

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_1 + a_2 x + a_3 y \\ a_4 + a_5 x + a_6 y \end{pmatrix},$$

where $(x, y)^T$ is the pixel location in the PanoramaVideo and $(x', y')^T$ is the corresponding location in the other PanoramaVideo. The cross-correlation coefficients (ρ) of the overlapping parts of scene spaces are calculated. The ρ is expressed as,

$$\rho = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} I(x_i, y_j) I'(x'_i, y'_j)}{\sqrt{\left(\sum_{i=0}^{n}\sum_{j=0}^{m} I(x_i, y_j)^2 - \left(\sum_{i=0}^{n}\sum_{j=0}^{m} I(x_i, y_j)\right)^2\right)}} \times \frac{}{\left(\sum_{i=0}^{n}\sum_{j=0}^{m} I'(x'_i, y'_j)^2 - \left(\sum_{i=0}^{n}\sum_{j=0}^{m} I'(x'_i, y'_j)\right)^2\right)}$$

where $I(x_i, y_j)$ denotes the image intensity at location (x_i, y_j), and $I'(x'_i, y'_j)$ denotes the image intensity at location (x'_i, y'_j). We match sev-

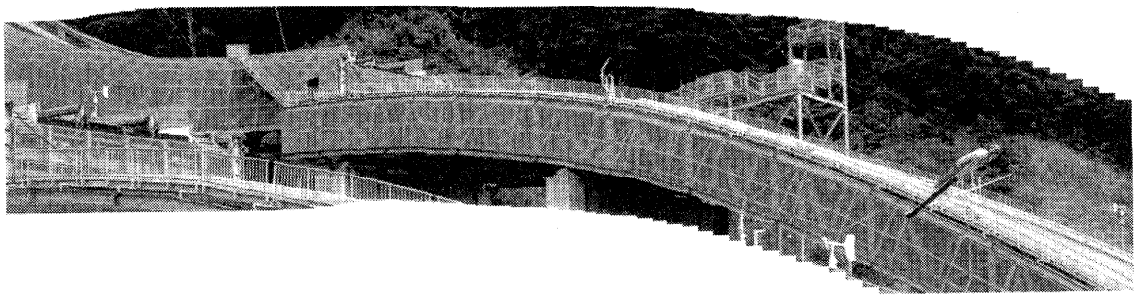

Fig. 5 The Panorama Video produced from shot captured with pan and tilt camera operations.

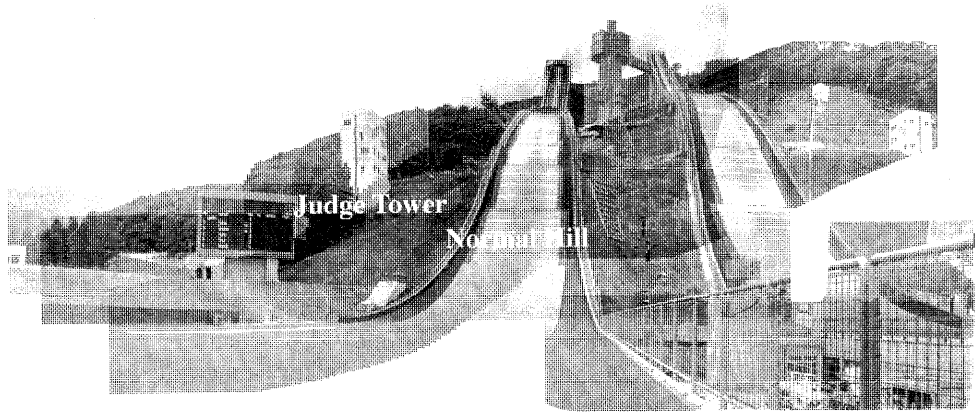

Fig. 6 A spatial synthesized scene space created from 9 shots.

eral points in scene space, and fit into the Affine parameters by the least squares method.

If we watch each shot individually, we can not understand the spatial relationship of the objects captured in each shot. The spatial synthesized scene space allows us to grasp the spatial object structure in terms of position and size. **Figure 6** shows a sample of the spatial synthesized scene space created from nine shots. When we watch each shot separately, we can not easily understand the spatial relationship between the "normal hill" and "judge tower". However, if we see the synthesized scene space, we can easily understand them.

4.1.2 Object Stamping on Scene Space

We used a moving object and trajectory extraction method based on the frame-difference (approximate background difference) technique. In the extraction method, first, we transform video frames to no-camera operation video frames to take away camera operations. Let $F(x, y, t)$ represent the no-camera operation video frame at time t. We get difference data between $F(x, y, t)$ and $F(x, y, t + \delta t)$. The δt denotes an interval of frames. The frame difference data is expressed as,

$$D(x, y, t) = |F(x, y, t + \delta t) - F(x, y, t)|.$$

And we get $H(x, y, t)$ from $D(x, y, t)$ and $D(x, y, t + \delta t)$, as the following,

$H(x, y, t) = D(x, y, t)$,
if $D(x, y, t)$ is larger than $D(x, y, t + \delta t)$,
$H(x, y, t) = D(x, y, t + \delta t)$,
if $D(x, y, t)$ is smaller than $D(x, y, t + \delta t)$.

The moving object target data, $T(x, y, t)$, to be extracted is enhanced by binarizing the $H(x, y, t)$. In the method of removing noise and integrate scattered target area, the $H(x, y, t)$ is blurred by Gaussian filter repeatedly. We binarize the filtered $H(x, y, t)$ and then label the binarized area. We extract feature (a color histogram) from each labeled area, and select a target object area. The similarity value to extract object is calculated by the following equation,

$$S = \frac{\sum_{k=-K}^{K} \min(F(k)_t, F_{t+1}(k))}{\sum_{k=-K}^{K} F_t(k)},$$

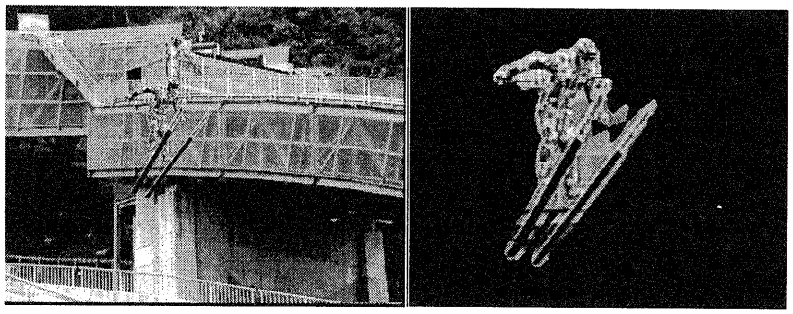

Fig. 7 A result of object extraction.

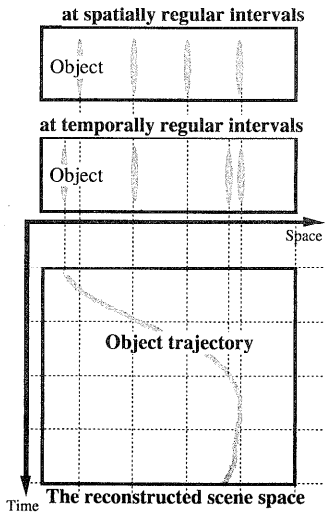

Fig. 8 Two types of Stroboscopic PanoramaVideo.

where F_t is the color histogram at time t. The K denotes a dynamic range of the histogram. The feature in first frame is calculated the object area which is given by user. A sample of an extracted object is shown in **Fig. 7**. We fix the extracted objects in a reconstructed scene space to represent motion object. We call this Stroboscopic PanoramaVideo. Stroboscopic PanoramaVideo has two types of representation as shown in **Fig. 8**, 1) fixed at spatially regular intervals, 2) fixed at temporally regular intervals.

Stroboscopic PanoramaVideo allows us to understand not only the object's absolute motion but also changes in the object's form. By using object information and the relationship between shots, we can overlap the Stroboscopic PanoramaVideo using fixed, spatially regular intervals. **Figure 9** shows the overlapped Stroboscopic PanoramaVideo. We used two shots of different skiers. This representation allows us to understand the spatiotemporal difference between theirs forms intuitively. Watching the two shots on different monitors is not as effective.

4.2 Space Oriented Interaction

In this paper, we propose a space oriented video interface with direct access to spatiotemporal video contents. The proposed interface uses a panoramic representation to provide access cues; time code access is not used. The video access mainly consists of two types of operations: 1) selecting the access point and 2) manipulating (play, stop and so on) the video. The space-oriented visualization mentioned in Section 4.1 is effective for selecting the manipulation points on the video directly. If you click on a specific part of the panoramic representation, the video will be replayed from the point selected. Spatially accessing the panoramic view equals temporally accessing the video sequence.

In the real world, we try to understand some events by using various viewing styles. We assume that there are three viewing styles: overview, place-based point-view, and motion-based point-view. The overview stage provides a view of the whole space. In the point-view stage, we watch and concentrate on the events happening. We use the place-based point-view when we want to pay attention to a specific place and observe what is happening. When we focus on a moving object, we utilize the motion-based point-view. We get information from the real world while going back and forth between these stages.

Figure 10 shows our implementations of the space oriented play back modes corresponding to the above viewing styles: 1) panoramic playback, 2) place-based playback and 3) motion-

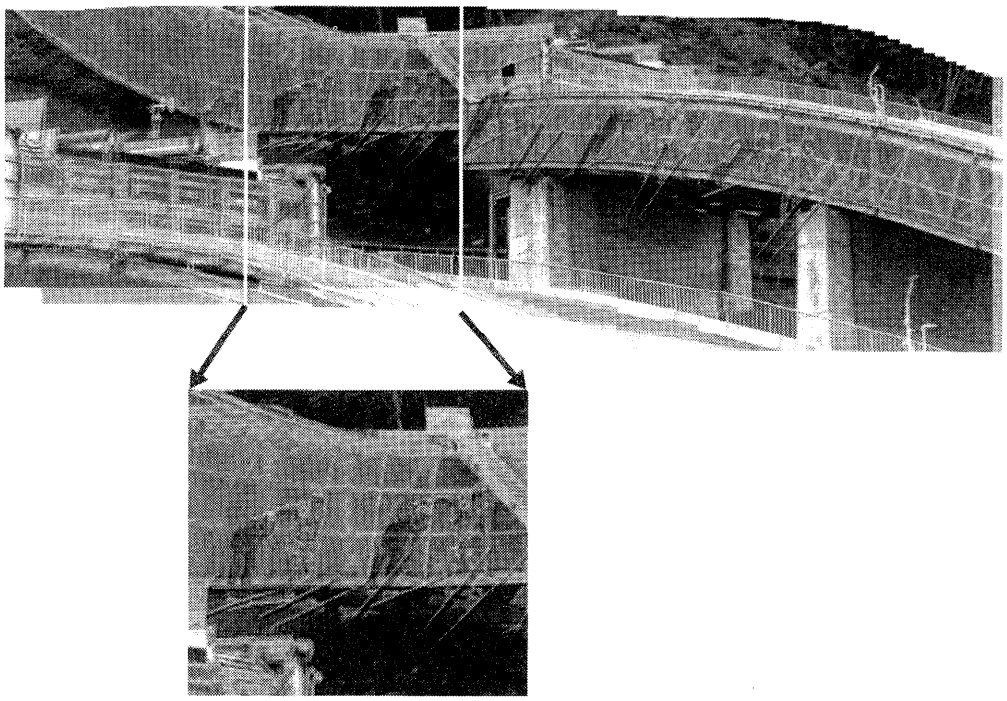

Fig. 9 The overlapped Stroboscopic PanoramaVideo.

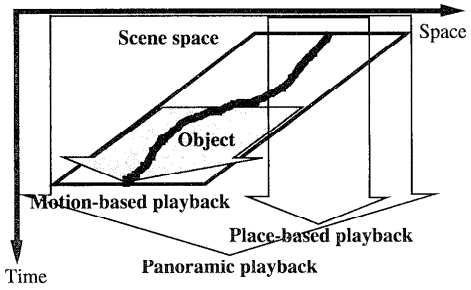

Fig. 10 Space-based interaction modes.

based playback. The panoramic playback mode can replay a shot using the panoramic scene space. We can watch the moving objects as if we were there. If we want to watch some part of the scene space in which some events are happening, the place-based playback mode supports us in accessing that spatial part directly. Playback is performed only for the place of interest. The motion-based playback mode replays the momentary object motion selected by the user by clicking on the fixed object in the scene space. This mode is effective in understanding the occurrence of temporal events in the shot.

The proposed video interface offers both spatial and temporal manipulation. If you "touch" some part of the panoramic spot and move it, the video will be replayed at the speed matching your movement. This provides us with seamless and/or intuitive manipulation (playback, stop, forward, rewind, etc.) (**Fig. 11**).

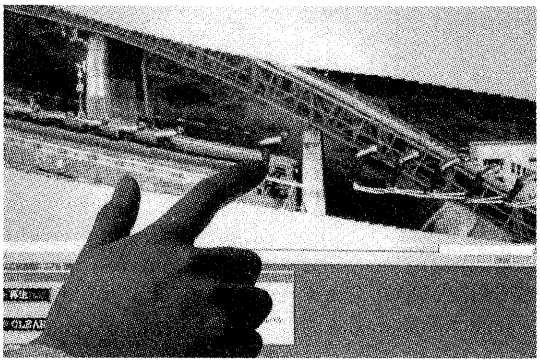

Fig. 11 The VideoStyler with direct manipulation.

We usually interact with a video to get spatiotemporal information for various purposes, for example, editing, retrieving, entertainment and so on. The best video interface should have various levels of granularity to support different interaction styles and intentions. When we

watch TV, video movies, etc., we are satisfied with rough granularity (for example title, action scene, music scene, etc.) in terms of interaction. The proposed VideoBrowser provides many more levels of interaction granularity. Video creators and/or editors, sports coaches and/or commentators, etc. require fine grain interaction. To edit videos, individual frames should be handled, for example "In point" and "Out point" selection. To analyze captured motion, larger frame groups are needed. PanoramaVideo and VideoJigsaw allow us to interact at the shot and/or frame level and so provides seamless fine grain interaction. This allows the user to catch and to handle the important action points from the shots effectively and intuitively.

5. Conclusion

We proposed VideoStyler, a new video interface made possible by multi-dimensional video computing. VideoStyler realizes spatiotemporal visualizing and interacting; the video contents can be handled as the user's wants. We introduced the video processing framework of "multi-dimensional video computing" for analyzing videos, creating new structures, and restyling and visualizing video content. Our discussion focused on two types of video structures that match the user's purpose and the kinds of video. Visualization and interaction processes based on these structures were also discussed. The proposed space oriented visualization and interaction realizes panoramic representation, synthetic representation, and direct access without using time code. We implemented these functions in a video interface and discussed the efficiency of the proposed interface for grasping and understanding video content directly and intuitively. We believe our ideas for various video browsing and new intuitive space oriented video accessing interfaces are very promising for future video usage and stimulating applications.

Acknowledgments We are grateful to Dr. Yukio Tokunaga, Executive Manager of the Advanced Video Processing Laboratory, NTT Human Interface Laboratories, for his encouragement during this research. We would also like to thank our colleagues for their helpful advice and discussions.

References

1) Brondmo, H.P. and Davenport, G.: Creating and viewing the Elastic Charles – A hypermedia journal, *Hypertext: State of the Art*, McAleese, R. and Green, C.(Eds.), pp.43–51, Ablex (1990).
2) Tonomura, Y. and Abe, S.: Content-oriented Visual Interface using Video Icon for Visual Database Systems, *J. Visual Languages and Computing*, Vol.1, No.2, pp.183–198 (1990).
3) Ueda, H., Miyatake, T. and Yoshizawa, S.: IMPACT: An Interactive Natural-Motion-Picture Dedicated Multimedia Authoring System, *Proc. CHI '91*, pp.343–350 (1991).
4) Mill, M., Cohen, J. and Wong, Y.Y.: A Magnifier Tool for Video Data, *Proc. CHI '92*, pp.93–98 (1992).
5) Elliott, E. and Davenport, G.: Video Streamer, *Proc. CHI '94*, pp.65–66 (1994).
6) Tonomura, Y., Akutsu, A., Otsuji, K. and Sadakata, T.: VideoMAP and VideoSpaceIcon: Tools for Anatomizing Video Content, *Proc. INTERCHI '93*, pp.131–138 (1993).
7) Teodosio, L. and Bender, W.: Salient Video Stills: Content and Context Preserved, *Proc. ACM Multimedia '93*, pp.39–46 (1993).
8) Arman, F., Depommier, R., Hsu, A. and Chiu, M.-Y.: Content-based Browsing of Video Sequences, *Proc. ACM Multimedia 94*, pp.97–103 (1994).
9) Kelly, P.H., Katkere, A., Kuramura, D.Y., Moezzi, S., Chatterjee, S. and Jain, R.: An Architecture for Multiple Perspective Interactive Video, *Proc. ACM Multimedia '95*, pp.201–212 (1995).
10) Kanade, T., Narayanan, P.J. and Rander, P.W.: Virtualized Reality: Concept and Early Results, IEEE Workshop on the Representation of Visual Scenes, Boston (June 1995).
11) Otsuji, K., Tonomura, Y. and Ohba, Y.: Video Browsing using Brightness Data, *Proc. SPIE, Visual Communication and Image Processing 91*, Vol.1606, pp.980–989 (1991).
12) Nagasaka, A. and Tanaka, Y.: Automatic Video Indexing and Full-Video Search for Object Appearances, *Video Database Systems*, II, Kunth, E. and Wegner, L.M. (Eds), pp.113–127, Elsevier Science, North-Holland (1992).
13) Zhang, H.J., Kankanhalli, A. and Smoliar, S.W.: Automatic Partitioning of Full-motion Video, *Multimedia Systems*, Vol.1, pp.10–28 (1993).
14) Yeung, M.M. and Liu, B.: Efficient Matching and Clustering of Video Shots, *Proc. ICIP '95*, Vol.1, pp.338–341 (1995).
15) Hoter, M.: Differential estimation of the global

motion parameters zoom and pan, *Signal Processing*, Vol.16, No.3, pp.249–265 (1989).
16) Akutsu, A. and Tonomura, Y.: Video Tomography: An efficient method for Camerawork Extraction and Motion Analysis, *Proc. ACM Multimedia '94*, pp.349–3566 (1994).
17) Irani, M. and Peleg, S.: Motion Analysis for image enhancement: Resolution, occlusion and transparency, *J. Visual Communication and Image Representation*, Vol.4, No. 4, pp.324–335 (1993).
18) Mann, S. and Picard, R.W.: Virtual Bellows: Constructing high quality stills from video, *Proc. ICIP '94*, Vol.I, pp.363–367 (1994).
19) Sawhney, H., Ayer, S. and Gorikani, M.: Model-based 2D and 3D dominant motion estimation for Mosaicking and video representation, Technical Report, IBM Almaden Res. Ctr. (1994).
20) Brown, G. and Cooke, M.: Computational Auditory Scene Analysis, *Computer Speech and Language*, Vol.8, pp.297–336 (1994).
21) Pfeiffer, S., Fischer, S. and Effelsberg, W.: Automatic Audio Content Analysis, *Proc. ACM Multimedia '96*, pp.21–30 (1996).
22) Minami, K., Akutsu, A., Hamada, H. and Tonomura, Y.: Enhanced Video Handling based on Audio Analysis, *Proc. IEEE Multimedia Computing and Systems '97*, pp.219–226 (1997).
23) Tonomura, Y., Akutsu, A., Taniguchi, Y. and Suzuki, G.: Structured Video Computing, IEEE Multimedia, Vol.1, No.3, pp.34–43 (1994).
24) Taniguchi, Y., Akutsu, A., Tonomura, Y. and Hamada, H.: An intuitive and efficient access interface to real-time incoming video based on automatic indexing, *Proc. ACM Multimedia '95*, pp.25–33 (1995).
25) Akutsu, A., Tonomura, Y. and Hamada, H.: VideoStyler: Multi-dimensional video computing for eloquent media interface, *Proc. ICIP 95*, Vol.1, No.1, pp.330–333 (1995).
26) Akutsu, A., Tonomura, Y. and Hamada, H.: Space-based Visualization and Access of Video Contents, *Third Joint Workshop on Multimedia Communications*, pp.1-2-1–1-2-8, Korea (Oct. 1996).

(Received June 30, 1997)
(Accepted March 6, 1998)

Akihito Akutsu is a research engineer in the Advanced Video Processing Laboratory at NTT Human Interface Laboratories. He researches video-handling techniques based on image processing. He received his BE and MS degrees in image science and engineering from Chiba University in 1988 and 1990.

Yoshinobu Tonomura is a research group leader in the Advanced Video Processing Laboratory at NTT Human Interface Laboratories. He researches visual communication systems. He was a visiting researcher at the Media Laboratory at the Massachusetts Institute of Technology in 1987-1988. He is working on video-handling techniques. Tonomura received his BE and MS degrees in electronic engineering from Kyoto University in 1979 and 1981.

Hiroshi Hamada is a executive manager of NTT Customer Equipment Department. His research interests include multimedia user interfaces, speech processing, and human factors. He received his BE and MS degrees in electrical engineering, and PhD degree in computer science, from University of Electro-Communications in 1978, 1980, and 1996, respectively.