INVITED PAPER *Excellent Paper Originally Published in Japanese*

# PanoramaExcerpts: Video Cataloging by Automatic Synthesis and Layout of Panoramic Images*

**Yukinobu TANIGUCHI**†, **Akihito AKUTSU**††, *and* **Yoshinobu TONOMURA**††, ***Regular Members***

**SUMMARY** Browsing is an important function supporting efficient access to relevant information in video archives. In this paper, we present PanoramaExcerpts — a video browsing interface that shows a catalogue of two types of video icons: panoramic and keyframe icons. A panoramic icon is automatically synthesized from a video segment taken with camera pan or tilt using a camera parameter estimation technique. One keyframe icon is extracted for each shot to supplement the panoramic icons. A panoramic icon represents the entire visible contents of a scene extended with a camera pan or tilt, which is difficult to represent using a single keyframe. A graphical representation, called camera-work trajectory, is also proposed to show the direction and the speed of camera operation. For the automatic generation of PanoramaExcerpts, we propose an approach to integrate the following: (a) a shot-change detection method; (b) a method for locating segments that contain smooth camera operations; (c) a layout method for packing icons in a space-efficient manner. In this paper, we mainly describe (b) and (c) with experimental results.

***key words:*** *video handling, video browsing video analysis, panoramic image, image mosaic*

## 1. Introduction

The amount of digital video information is rapidly increasing according to the development of digital video technologies and storage devices. The more video information is accumulated, the more difficult it becomes to find the desired information. A video browsing interface, which allows users to efficiently access videos, is an important component in digital video applications such as video libraries and nonlinear video editing systems.

One approach to generate a video browsing interface is to display a catalogue of keyframes [2] extracted from a video. The important issues are to extract keyframes that well represent the video contents, and to present the keyframes in such a way that users can see the temporal context and have an overview of the entire video.

Many methods have been proposed for selecting keyframes from a video. One common approach is to segment the video into shots and to select one or several keyframes from each shot [1], [10], [15]. A trivial method is to select the first frame of a shot as its keyframe. To select more representative keyframes, some researchers have proposed the use of motion analysis [11] and color features [23]. The repetition of shots was detected to reduce the redundancy of keyframes [5], [22].

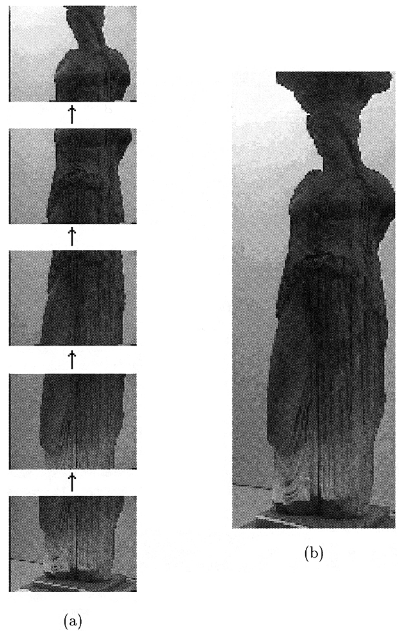


**Fig. 1** Limitation of the keyframe extraction approach: (a) input image sequence, (b) the panoramic image created from (a).

One limitation of these approaches is that it is not always possible to select keyframes that well represent the entire image contents. Figure 1 (a) shows an image sequence with a camera tilt; no single frame in the sequence can represent the entire contents.

The solution we propose is to synthesize a panoramic image by composing a sequence, as shown in Fig. 1 (b), and using it as the representative image. The panoramic image intuitively represents the entire contents of the sequence. Our approach is to *synthesize* representative images and not to *select* keyframes as in conventional methods. In this paper, we propose the video browsing interface, PanoramaExcerpts, which shows a combination of keyframe and panoramic icons, and the system to automate the generation.

Manuscript received June 20, 2000.

†The author is with NTT Cyber Solution Laboratories, Yokosuka-shi, 239–0847 Japan.

††The authors are with NTT Cyber Solution Laboratories, Kyoto-fu, 619–0237 Japan.

*This article was originally published in the Transactions of the Institute of Electronics, Information and Communication Engineers D-II, vol.J82-D-II, no.3, pp.390–398, 1999.

The use of panoramic images for video browsing has previously been mentioned [2], [7], [12], [17], [20]. Massey and Bender [7] applied panoramic images, which they called salient stills, to the generation of a comic book from a video. Their system requires user's manual inputs to select segments for constructing the salient stills, and is not suitable for video browsing. To our knowledge, none of the implemented systems thus far fully automate the process of enumerating and synthesizing panoramic images from a video.

Panoramic image synthesis, also known as image mosaicing, has been applied to virtual reality [18], video compression [4], video reuse [7], 3D scene reconstruction, and moving object recognition [14]. Many techniques have been developed [3], [9], [17]; however, most existing methods aim at generating high-resolution panoramic images, and assume as input a "contiguous image sequence taken in a controlled environment." In contrast, we deal with "long edited videos." The problem is that edited videos contain noise, such as object motion and camera vibrations; this means that the estimated camera parameters are not always reliable for all shots. To automatically extract visually good panoramic images, we must first detect smooth camera operations, as distinct from camera vibrations. Smith et al. [13] used motion vectors to detect and classify camera operations.

The integration of panoramic icons with video browsing interfaces leads to another problem: panoramic icons have irregular shapes according to camera operation, while keyframe icons are all rectangular and of the same size. To generate a compact catalogue view, we also propose a method for packing icons while maintaining temporal order.

This paper is organized as follows: Sect. 2 gives an overview of the PanoramaExcerpts system. Sections 3, 4, and 5 propose methods for generating panoramic icons, for visualizing the speed and direction of camera operations, and for packing icons in a window, respectively. The experimental results are also reported. Section 6 discusses the PanoramaExcerpts system in terms of interface and processing efficiency.

## 2. PanoramaExcerpts Generation Process

The generation of PanoramaExcerpts consists of the following steps (Fig. 2):

1. Shot-change detection: An input video is segmented into shots by detecting shot changes such as cuts and dissolves.
2. Camera parameter estimation: For each shot, the camera parameters are estimated.
3. Detecting stable camera operations: Using the estimated camera parameters, we detect segments from which panoramic icons can be constructed; these segments contain stable camera pan, tilt, and zoom operations.
4. Panoramic icon synthesis: For each stable camera operation segment, a panoramic icon is constructed by composing the image sequence in the segment.
5. Keyframe extraction: For each shot with no stable camera operation, a keyframe icon is extracted — currently the first frame of the shot.
6. Icon layout: The panoramic and keyframe icons are laid out in a window so that screen space is efficiently used while maintaining temporal order.

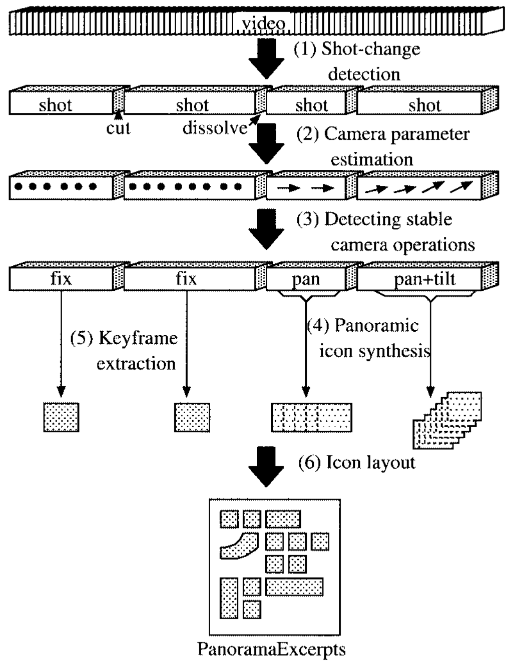



**Fig. 2** Overview of the PanoramaExcerpts system.

The distinct steps in the proposed system are 3 and 6. Step 3 is important to extract visually good panoramas. Step 6 is needed to generate compact catalogue views. Sections 3 and 5 describe steps 2 through 4 and step 6, respectively.

Step 1 is implemented using the method described in ref. [16]. The method detects shot changes including dissolves and fades.

## 3. Generating Panoramic Icons

The process of synthesizing panoramic icons consists of the following four steps: camera parameter estimation, detecting stable camera operations, camera operation classification, and panoramic icon synthesis. The following sections detail these steps.

### 3.1 Camera Parameter Estimation

Let $f(x, y)$ and $f'(x, y)$ be two consecutive images in a video. The camera model used here is

$$(x', y') = (ax, ay) + (d_x, d_y), \tag{1}$$

where $(x, y)$ and $(x', y')$ are corresponding points in images $f$ and $f'$, respectively. This model accounts for frequently used camera operations — pan, tilt, and zoom; parameters $a$, $d_x$, and $d_y$ correspond to zoom, pan, and tilt, respectively, where the origin of the coordinate system is the center of the image. Equation (1) approximately models camera pan, tilt, and zoom operations under the condition that the focal length is sufficiently large. We use this simple camera model because we place greater importance on computational efficiency than the accuracy of the estimated parameters.

To compute the dissimilarity between images $f$ and $f'$, the mean square error (MSE) is defined as follows:

$$MSE(\theta) = \frac{1}{N} \sum_{(x,y)} \{f(x, y) - f'(x', y')\}^2, \tag{2}$$

where $\theta = (a, d_x, d_y)$, the summation is taken for all overlapping pixels, and $N$ is the number of pixels†. The parameter $\theta$ that minimizes $MSE(\theta)$ is found using the coarse-to-find strategy [4] for efficient minimization: Image pyramids are constructed for frames $f$ and $f'$. Low-resolution images are used to find the coarse estimate, and then higher resolution images are utilized to improve the accuracy.

### 3.2 Detecting Stable Camera Operations

We detect a stable camera operation segment when time interval $I = [s, e]$ satisfies the following three conditions:

- Duration: a camera operation lasts for more than a specified period of time $D$, i.e., $e - s \geq D$;
- Smoothness: The camera operation is smooth, i.e., $\hat{\theta}_t \cdot \hat{\theta}_{t-1} > 0,\ t \in I$, where $\hat{\theta}_t$ is the camera parameter estimated at time $t$;
- Goodness of fit: $MSE$ minimized by the estimated parameter $\hat{\theta}_t$ is sufficiently smaller than that without minimization; specifically,

  $$MSE(\hat{\theta}_t)/MSE(\mathbf{0}) < T,\ \ t \in I, \tag{3}$$

  where $T$ is a threshold, and $0 \leq T \leq 1$.

### 3.3 Camera Operation Classification

Detected stable camera operation segments are classified into pan, tilt, zoom, or a combination of these. The amount of translation and zoom in a segment are used in the classification. When the total amount of translation along the $x$ direction exceeds a certain threshold, it is decided that the segment includes camera pan.

### 3.4 Panoramic Icon Synthesis

A panoramic image is synthesized for each stable camera operation segment. A seamless panoramic image is composed by overlapping the image sequence after the images are translated and scaled according to equation (1), and the estimated camera parameters are substituted. Overlapping pixels are overwritten by the latter image. A zooming image sequence is spatially scaled in such a way that the image showing the widest angle of view is of a predefined size, so that all the resulting icons have similar sizes.

### 3.5 Experimental Results

The number of stable camera operations detected by the proposed method is compared with the number of manually detected ones, and the result is shown in Table 1. We investigated only camera operations that last more than one second. The precision $P$ and recall $R$ measures are defined as:

$$P = (T - M - F)/(T - M),\ \ R = (T - M)/T,$$

where $T$, $M$, and $F$ are the numbers of stable camera operations, missed detections, and false positives, respectively. A combination of panning and tilting is counted as both a pan and a tilt. When a stable camera operation is detected as two camera operations, it is regarded as a false positive. As shown in Table 1, the proposed method can detect over 90% of stable camera operations in terms of the precision measure. A camera operation segment is missed when the camera follows an object that occupies a large portion of the frame. The recall rate of zoom operations is low compared with those of pans and tilts, because the change caused by zoom is so slight that the goodness-of-fit condition is not always met.

The goodness-of-fit condition (3) was found to be useful in reducing false detection. Without this condition, eight additional false positives were detected for the video data "Greece."

The extraction of panoramic icons failed in the following situations. Two camera operation segments were sometimes detected as one because the boundary was not detected as a shot change; Fig. 3 (a) shows a panoramic image wrongly synthesized by combining two shots. Conversely, it was sometimes observed that one camera operation segment was divided into two, because the continuity of operation was wrongly judged; this produced two panoramic icons from a single camera operation. It seems preferable to adjust threshold

†More specifically, for each pixel $(x, y)$ in image $f$, the corresponding pixel $(x', y')$ is calculated with equation (1); if pixel point $(x', y')$ is in image $f'$, the summation is taken with equation (2), otherwise the pixel is ignored. $N$ is the number of pixels summed.

**Table 1** Precision $P$ and recall $R$ measures of the camera operation classification method. $T$, $M$, and $F$ denote the numbers of camera operations, false negatives, and false positives, respectively.

| source | pan | | | tilt | | | zoom | | |
|---|---|---|---|---|---|---|---|---|---|
| | $T$ | $M(R)$ | $F(P)$ | $T$ | $M(R)$ | $F(P)$ | $T$ | $M(R)$ | $F(P)$ |
| Greece | 33 | 2(94%) | 0(100%) | 18 | 0(100%) | 0(100%) | 6 | 2(67%) | 0(100%) |
| CNN | 8 | 2(75%) | 0(100%) | 5 | 0(100%) | 0(100%) | 11 | 2(82%) | 2(82%) |
| India | 24 | 0(100%) | 2(92%) | 17 | 2(88%) | 1(94%) | 12 | 6(50%) | 0(100%) |

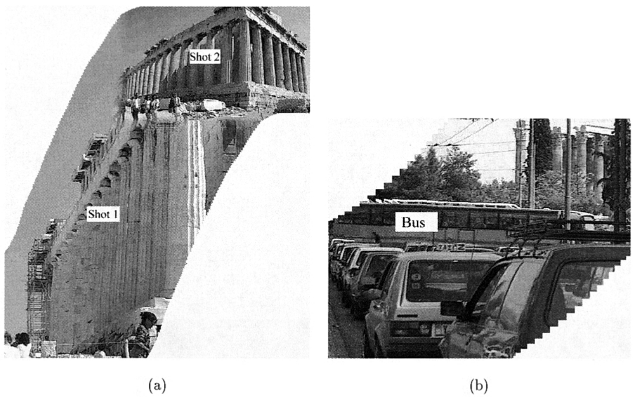



**Fig. 3** Examples of inappropriate panoramic icons: (a) a panoramic icon wrongly generated by combining two consecutive shots, (b) a panoramic icon in which the bus is longer than an actual one, due to motion.

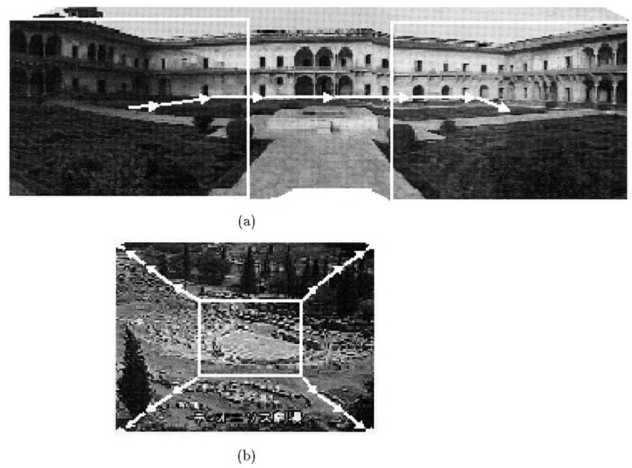



**Fig. 4** Visualization of camera operations: (a) for shots without zoom operation, and (b) for shots with zoom operation.

$T$ so that a high precision ratio may be obtained, as false positives have a more negative effect on the system than false negatives.

Object motion may cause the following problems: (i) estimated camera parameters could be erroneous, and (ii) synthesized panoramic images could be distorted. In terms of (i), due to the goodness-of-fit condition (3), camera operations containing significant object motion were judged unstable, and panoramic icons were not synthesized. In terms of (ii), the distortion was sometimes strikingly visible, particularly when the object was moving across the frame boundary. Figure 3 (b) shows another example of a wrongly generated panoramic icon, in which the bus is shown longer than an actual one because the moving bus is on the boundary of the frame.

## 4. Camera-Work Trajectory

Panoramic icons synthesized as described in the previous sections indicate the type of camera operation by their shapes; however, the direction and the speed are not represented. It is important to represent this information as an aid in finding video footages for video editing. In this section, we propose a graphical representation, called camera-work trajectory, for intuitively visualizing information related to camera operations obtained in the camera parameter estimation step.

The PanoramaExcerpts system switches visualization schemes depending on the classification result of a camera operation. For camera operation segments without zooming, as shown in Fig. 4 (a), an arrow originating at the center of an image indicates the direction and speed of the camera pan. For zooming segments, as in Fig. 4 (b), arrows originating at the four corners of the frame are used to represent the scale change.

There is little work available on the visual representation of camera-work attributes. VideoSpaceIcon [20] uses a 3D icon to show the camera-work attributes, and Ueda et al.[21] used a diagram drawn on keyframe images.

The novel point of our method is that it represents both camera operation information and visual contents; panoramic icons show visual contents taken with camera operations.

## 5. Icon Packing

Many techniques have been developed for laying out icons for efficient video browsing: simply displaying multiple icons in a window [19], [21], VideoMagnifier [8] which provides hierarchical views, and VideoPoster [22] which controls the size of icons depending on their relevance. These techniques allow the removal or the resizing of icons for a compact browsing interface.

In this section, we propose a method for packing icons within a two-dimensional region (e.g., in a window) in a space-efficient manner; we do not deal with layouts that allow resizing and removal. Panoramic

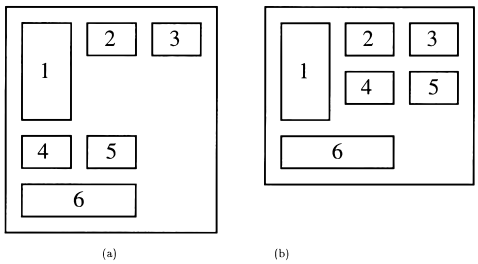



**Fig. 5** Layout generated by (a) method A, and (b) our proposed method.

icons have different shapes and sizes depending on the camera operation represented, so that effective icon packing is necessary for efficient browsing.

A naive layout method, called method A, which places icons from left to right in multiple lines, produces large dead spaces as shown in Fig. 5 (a).

Our proposed method allows a line to be subdivided into multiple rows, as illustrated in Fig. 5 (b) for efficient space utilization.

The packing problem in general has been well studied [6], and many methods have been developed. They have primarily been developed for non-interactive applications, such as VLSI design, and not suitable for our purpose. Our requirements are (i) to preserve the temporal order of icons, and (ii) to be sufficiently computationally efficient for interactive use.

### 5.1 Proposed Method

In this subsection, we describe the detailed procedure for packing icons. Let us consider rectangular pieces $R_i$, $i = 1, \ldots, N$, each of which corresponds to the bounding box of an icon; $N$ is the number of icons, and the edges of $R_i$ are parallel to those of the window. For simplicity, we consider the problem of packing $R_i$ in a window whose width is fixed $W$ and whose height is unlimited. We place icons from the top-left corner under the condition that temporal order is preserved.

The procedure is as follows:

1. Initialize occupied region $O = \phi$. Occupied region $O$ has a step-like shape as shown in Fig. 6, and thus can be specified by its step corners $s_j = (x_j, y_j)$, $j = 1, \ldots, N_s$. Note that the origin of the coordinate system is the top-left corner of the window. Initially, $(x_1, y_1) = (0, 0)$, and $N_s = 1$.
2. For each piece $R_i$, $i = 1, 2, \ldots, N$:
   - 2.1. For each step corner$(x_j, y_j)$, $j = 1, 2, \ldots, N_s$: Let $D_j$ be the region of dead space produced by placing $R_i$ justified at the $j$-th step corner $(x_j, y_j)$, as depicted in Fig. 6. $D_j$ consists of two separate regions, where the subsequent pieces are prohibited from overlapping. If there is not sufficient space to place $R_i$, i.e., if $W(R_i) > W - x_j$, then let $D_j = \infty$, where $W(R_i)$ denotes the width of $R_i$.
   - 2.2. Find the placement that minimizes $D_j$, i.e., $j^* = \arg\min_j |D_j|$, where $|D_j|$ denotes the area of dead space $D_j$;
   - 2.3. Place $R_i$ justified at the $j^*$-th step corner;
   - 2.4. Update occupied region $O = O \cup R_i \cup D_{j^*}$, and the corresponding step corners $(x_j, y_j)$ and $N_s$. Return to step 2.1.

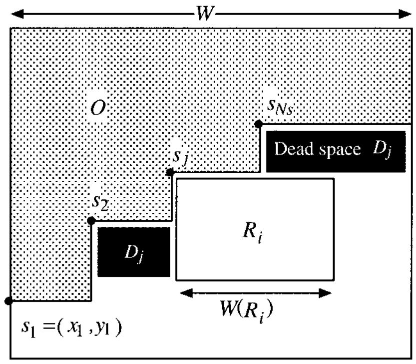



**Fig. 6** Dead space $D_i$ generated by placing rectangle $R_i$ justified at the $j$-th step corner $(x_j, y_j)$.

### 5.2 Experimental Results

To compare our method with method A, we define the packing ratio $PR = H(\text{proposed})/H(\text{method A})$, where $H$(proposed) and $H$(method A) denote the heights of the layouts as generated by our proposed method and that of method A, respectively. Packing ratios below one indicate that our method is better than method A. Note that our method is not assured of yielding optimal results; our method may provide a worse layout than method A[†].

First, we tested our method on randomly generated data sets. We generated 100 data sets of 500 pieces, whose width and height were randomly selected from 0 to 100; the packing ratio was measured for each data set by varying the window width from 100 to 600. The experimental results are shown in Fig. 7 (a). The vertical axis corresponds to the packing ratio. Our method yielded better results than method A for all data sets regardless of the width of the window.

Next, we tested our method on a real data set, containing 131 keyframe icons, 80 by 60 pixels in size, and 47 panoramic icons of different extents. The results are shown in Fig. 7 (b). As the window width increased, the packing ratio decreased to about 0.85. This means that about 15% of the screen space was saved using our method.

[†]For instance, the packing ratio exceeds one when it packs two rectangles $R_1$ (width=2, height=1) and $R_2$(width=1, height=2) in a window (width=3).

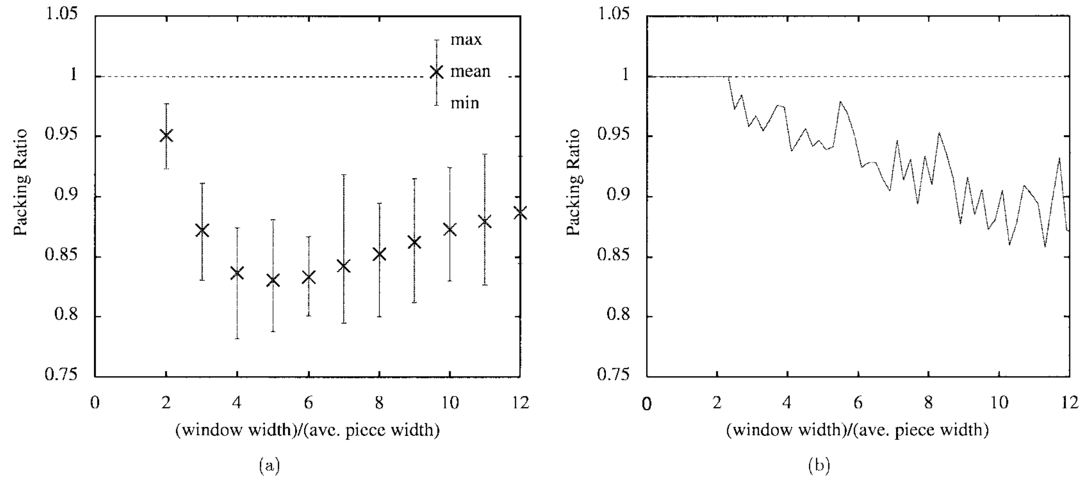



**Fig. 7** Packing ratio measured on: (a) randomly generated data sets, and (b) a real data set.

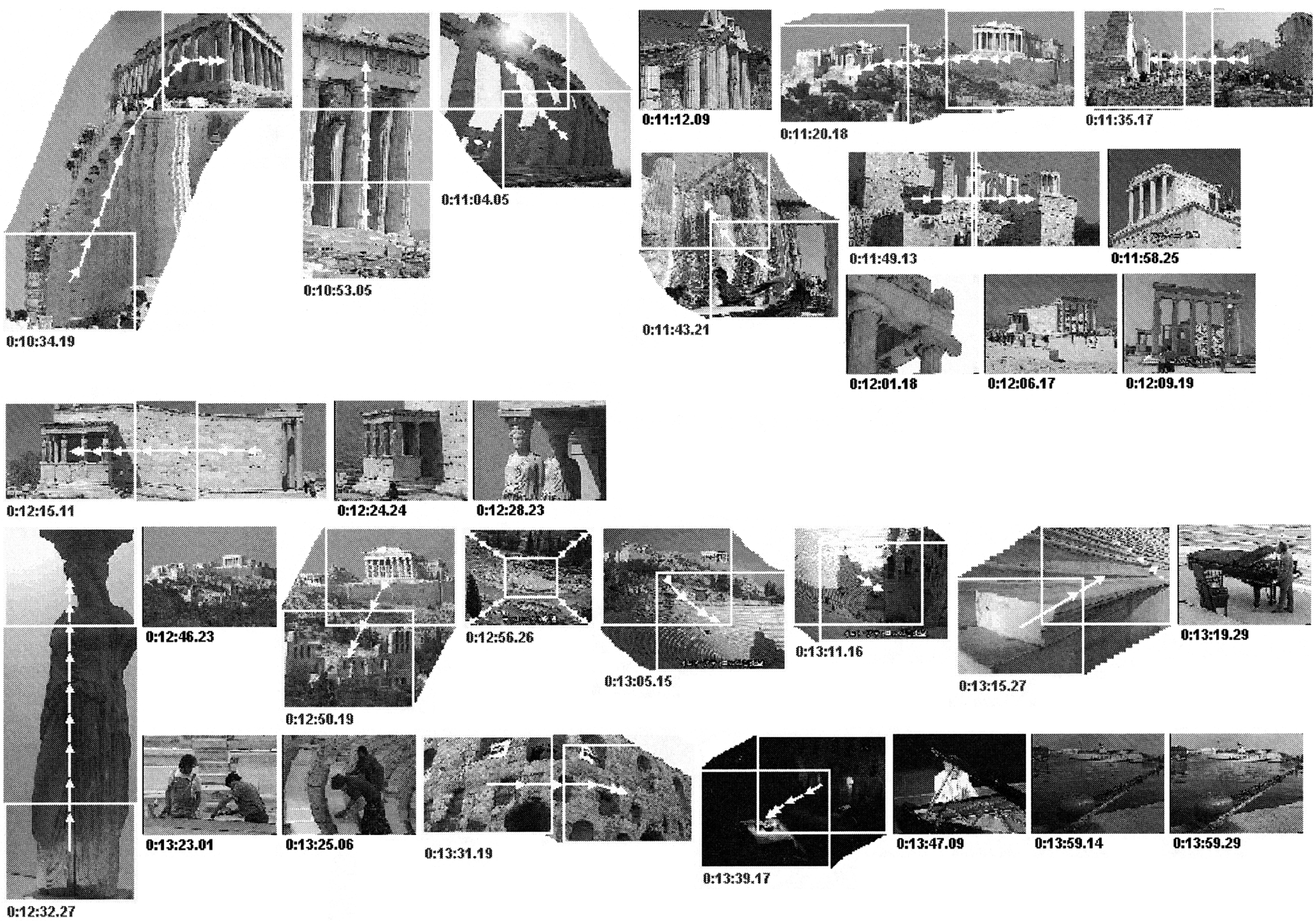



**Fig. 8** PanoramaExcerpts interface.

## 6. Discussion

### 6.1 PanoramaExcerpts Interface

Figure 8 shows an example of a PanoramaExcerpts interface generated from a travelogue on Greece. The PanoramaExcerpts interface has the following features:

- a panoramic icon efficiently represents the entire visible contents of the shot;
- a panoramic icon represents the type, direction,

and speed of camera operation with shape and camera-work trajectory.
- the succession of icons shows the global structure of the video, in other words, the context.

The effectiveness of the PanoramaExcerpts interface depends on the category of target videos. We generated PanoramaExcerpts for a variety of videos. For categories of video that contain many camera operations, such as travelogues and dramas, the PanoramaExcerpts interface is very effective. The 15-minute travelogue, which consists of 180 shots, contains 43 camera pans and tilts. Our system correctly identified 41 out of 43 camera operations, and produced panoramic icons. For other genres of videos, such as commercials and news programs which contain fewer camera operations, the PanoramaExcerpts interface is almost equivalent to a conventional catalogue of only keyframe icons.

The temporal order of icons was easily recognizable in most cases. However, when there were many camera pans and tilts in the sequence, some difficulty might occur.

Our prototype system supports functions such as replaying a video segment by clicking on the icon, editing the configuration of icons, and deleting redundant icons.

The proposed PanoramaExcerpts can be regarded as a first step toward reconstruction from a completed video "storyboard," which is a communication tool commonly used in the video production process. The video storyboard is an ideal form of video summary, and if it could be automatically reconstructed, it would be useful for efficient video browsing.

### 6.2 Processing Time

The time needed for generating the PanoramaExcerpts was less than the 70% of the duration of the input video, when images of 160 by 120 pixels were processed at a rate of 10 frames per second with a Pentium II 300 MHz personal computer. The original video was encoded in Motion JPEG format. The prototype system can process video signals such as TV broadcast in real time, and can be applied for real-time video indexing.

The time needed for packing icons was short compared with that needed for redrawing images.

## 7. Conclusion

We proposed the PanoramaExcerpts interface and presented the methods for generating and packing panoramic icons. We described the results showing the effectiveness of the PanoramaExcerpts interface. The proposed system has been proved sufficiently accurate and efficient for practical use.

Topics for further research include the development of video analysis techniques, such as motion analysis, to extract more effective representative images.

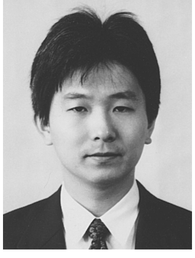
## Acknowledgment

The authors would like to thank Dr. Yukio Tokunaga, former executive manager of NTT Human Interface Laboratories, for his support of our research. Thanks are due to Dr. Hiroshi Hamada, Dr. Takashi Satou, Kenichi Minami, Yasuaki Niikura, and our colleagues for valuable discussions. All video images shown in Figs. 1,3,4,8 were reproduced by NTT Human Interface Laboratories with permission of the copyright holder Bungeishunju Ltd.

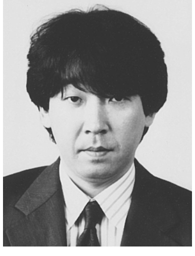
### References

[1] P. Aigrain and P. Joly, "The automatic real-time analysis of film editing and transition effects and its applications," Computer and Graphics, vol.18, no.1, pp.93–103, 1994.
[2] P. Aigrain, H.-J. Zhang, and D. Petkovic, "Content-based representation and retrieval of visual media: A state-of-the-art review," Multimedia Tools and Applications, vol.3, pp.179–202, 1996.
[3] A. Akutsu and Y. Tonomura, "Video tomography: An efficient method for camerawork extraction and motion analysis," Proc. ACM Multimedia '94, pp.349–356, ACM Press, 1994.
[4] P. Anandan, M. Irani, R. Kumar, and J. Bergen, "Video as an image data source: Efficient representations and applications," Proc. IEEE International Conference on Image Processing 95, pp.318–321, IEEE Computer Society Press, 1995.
[5] H. Aoki, S. Shimotuji, and O. Hori, "A shot classification method of selecting effective key-frames for video browsing," Proc. ACM Multimedia '96, pp.1–10, ACM Press, 1996.
[6] K.A. Dowsland and W.B. Dowsland, "Packing problems," European J. Operational Research, vol.56, pp.2–14, 1992.
[7] M. Massey and W. Bender, "Salient stills: Process and practice," IBM System Journal, vol.35, nos.3 and 4, pp.557–573, 1996.
[8] M. Mills, J. Cohen, and Y.Y. Wong, "A magnifier tool for video data," Proc. of CHI '92, pp.93–98, 1992.
[9] S. Mann and R.W. Picard, "Video orbits of the projective group: A simple approach to featureless estimation of parameters," IEEE Trans. Image Processing, vol.6, no.9, pp.1281–1295, 1997.
[10] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances," Proc. IFIP WG 2.6 2nd Working Conference on Visual Database Systems, pp.119–133, 1991.
[11] A. Pentland, R. Picard, G. Davenport, and K. Hasse, "Video and image semantics: Advanced tools for telecommunications," IEEE Multimedia, vol.1, no.2, pp.73–75, 1994.
[12] H.S. Sawhney and S. Ayer, "Compact representation of videos through dominant and multiple motion estimation," IEEE Trans. Pattern Anal. & Machi. Intelli., vol.18, no.8, pp.814–830, 1996.
[13] M.A. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding techniques," Carnegie Mellon University Technical Report, CMU-CS-97-111, 1997.

[14] R. Szeliski, "Video mosaics for virtual environments," Computer Graphics & Applications, pp.22–33, March 1996.
[15] Y. Taniguchi, Y. Tonomura, and H. Hamada, "A shot-change detection method and its applications to video access interfaces," IEICE Trans., vol.J79-D-II, no.4, pp.538–546, April 1996.
[16] Y. Taniguchi, A. Akutsu, and Y. Tonomura, "PanoramaExcerpts: Extracting and packing panoramas for video browsing," Proc. ACM Multimedia 97, pp.427–436, Addison-Wesley, 1997.
[17] L. Teodosio and W. Bender, "Salient video stills: Content and context preserved," Proc. ACM Multimedia '93, pp.39–46, 1993.
[18] L. Teodosio and M. Mills, "Panoramic overviews for navigating real-world scenes," Proc. ACM Multimedia '93, pp.359–364, 1993.
[19] Y. Tonomura and S. Abe, "Content oriented visual interface using video icons for visual database systems," J. Visual Languages and Computing, vol.1, pp.183–198, 1990.
[20] Y. Tonomura, A. Akutsu, K. Otsuji, and T. Sadakata, "Videomap and videospaceicon for anatomizing video content," Proc. INTERCHI '93, pp.131–138, IOS Press, 1993.
[21] H. Ueda, T. Miyatake, and S. Yoshizawa, "A proposal of an interactive motion picture editing scheme assisted by media recognition technology," IEICE Trans., vol.J75-D-II, no.2, pp.216–225, Feb. 1992.
[22] M.M. Yeung and B.-L. Yeo, "Video visualization for compact presentation and fast browsing of pictorial content," IEEE Trans. Circuits and Systems for Video Technology, vol.7, no.5, pp.771–785, 1997.
[23] H.J. Zhang, C.Y. Low, S.W. Smoliar, and J.H. Wu, "Video parsing, retrieval and browsing: An integrated and content-based solution," Proc. ACM Multimedia'95, pp.15–24, ACM Press, 1995.



**Yukinobu Taniguchi** received the B.E. and M.E. degrees in Mathematical Engineering from the University of Tokyo in 1990 and 1992, respectively. He is a research engineer in the Content Commerce Project, NTT Cyber Solution Laboratories. He received best paper award of the fifth ACM international multimedia conference. He is a member of Information Processing Society Japan and ACM.



**Akihito Akutsu** is a senior research engineer in the Intelligent Media Project, NTT Cyber Solution Laboratories. He received B.E. and M.S. degrees in image science and engineering from Chiba University in 1988 and 1990, respectively. He has been engaged in research on video handling techniques based on image processing.

**Yoshinobu Tonomura** is the project manager of the Intelligent Media Project, NTT Cyber Solution Laboratories. He has been engaged in research of visual communication systems. He was a visiting researcher at the Media Laboratory, MIT during 1987–1988. He is currently working on developing video applications. Tonomura received B.E. and M.S. degrees in electronics engineering from Kyoto University in 1979 and 1981, respectively.