An Exploratory Analysis Tool for a Long-Term Video from a Stationary Camera

Ryoji Nogami*, Buntarou Shizuki*, Hiroshi Hosobe†, and Jiro Tanaka*
* Department of Computer Science, University of Tsukuba, JAPAN
† National Institute of Informatics, JAPAN

***Abstract*—We present an interactive tool for the exploratory analysis of a long-term video from a stationary camera. The tool consists of three key methods: spatial change visualization, temporal change visualization, and similarity-based video retrieval. The first two methods summarize the long-term video, letting the user know where and when changes frequently occurred during a certain period, allowing the user to find an event of interest from the video. With the third method the user can search the video for a similar event, enabling the user to count events of interest and to observe distributions of such events. These methods are uniformly implemented using frame differences with 1-bit depth, making the implementation of these methods simple but efficient.**

***Keywords*—surveillance, similarity-based video retrieval, frame difference, omnidirectional camera, overhead video capture, spatiotemporal visualization, video summarization, video analytics, visual analytics.**

I. Introduction

Computer vision-based analysis of people's actions is used, for example, to examine customer stratums and flow lines in stores and to evaluate the effect of advertisements [1]–[4]. Typically the system automatically collects event data captured by a camera, such as the number and flow lines of people, and the user performs the analysis by observing the collected event data.

In contrast, we are interested in the exploratory analysis of a long-term video taken with a stationary camera. This exploratory approach enables the user to find events from his or her own viewpoint. However, for this purpose, the video replay speed needs to be sufficiently slow to enable the user to comprehend events. The analysis will be impractical if the amount of captured video equals a duration of several months.

In this paper, we present an interactive tool for the exploratory analysis of such a long-term video. Our tool provides three key methods for exploratory video analysis, namely, spatial change visualization, temporal change visualization, and similarity-based video retrieval. The first two methods, spatial change visualization and temporal change visualization, summarize the video by presenting visual information about where and when changes frequently occurred in the video during a certain period. With these two methods, the user can find an event of interest from the video. The third method, similarity-based video retrieval, allows the user to search the video for a similar event. These methods are uniform in the sense that all rely on frame differences. In our tool, a frame difference is a 1-bit monochrome image constructed from two successive video frames in such a way that the intensity of a pixel will be 1 if the corresponding pixels in the original frames are changed (with respect to a certain threshold) and 0 otherwise. The use of frame differences makes these methods simple but efficient.

Fig. 1. A video frame image taken with an omnidirectional camera mounted on the ceiling of a room in our laboratory.

We applied our tool to the analysis of an actual long-term video. We filmed the video using an omnidirectional camera mounted on the ceiling of a room in our laboratory, as shown in Figure 1. In this paper, we present a case study to demonstrate how our tool can be used to analyze long-term video.

II. Related Work

Video summarization has the potential to enable a user to interactively explore videos to detect events, both known and unknown in advance. Several researchers have studied video summarization in terms of the exploratory analysis of videos.

Summarizing a video by stacking the frames along the z-axis in a volumetric form has been investigated (e.g., [5] [6]–[8]). In this *summarization in a volumetric form*, the volume provides the user with an overview of the video and, at the same time, serves as an interface for seeking. This interface enables the user to browse the video and to interactively analyze each frame in detail. Aside from video summarization, [9] used a similar type of visualization in which a temporal series of ground-motion wave-field maps were stacked to form a volumetric form. The resulting volume showed the seismic wave propagation of an earthquake over time, allowing the user to observe the propagation at a glance. It also served as an interface for browsing the maps and analyzing each map.

Stacking feature changes between frames, or stacking recognition results of events/movements between frames, into one summarized image has also been investigated; by doing this,

the user should be able to gain an overview of a video at a glance (e.g., [10], [11]). [12] summarized a short piece of video (e.g., 30 minutes) by stacking the frames via alpha-blending, producing a similar representation. The combination of this *summarization by stacking* and summarization in a volumetric form has also been researched (e.g., [13]–[15]).

Because our focus is analyzing a long-term video from a stationary camera, we adopt two-dimensional summarization by stacking to enable the user to detect events for further analysis. The idea behind this adopt is that this technique enables the user to detect changes (events) in the video *by comparing* summaries of two different periods statically.

In the field of surveillance, frame differencing is used with background subtraction to detect moving objects (e.g., people walking) to track and/or to recognize them (e.g., [16], [17]). In contrast, we adopt a fairly simple and lightweight approach (i.e., the use of frame differences with 1-bit depth) to construct a video summarization. This is because our summarization is designed to provide the user with hints to know where, when, and how many changes (corresponding to differences between successive frames) have occurred in a long-term video, while allowing for user interpretation of the changes (including tracking and/or recognizing them).

Moreover, our tool provides a video summarization that utilizes similarity-based video retrieval, enabling the user to count events of interest (including anomalies) in a long-term video as well as to observe the distributions of such events.

III. Frame Difference-Based Approach

Here, we present our approach to analyzing a video taken with a stationary camera. Our approach is uniform in the sense that all key components of the analysis involve frame differences that are computed as differences between successive frames in the video. Specifically, the approach consists of three key components: spatial change visualization, temporal change visualization, and similarity-based video retrieval.

A. Spatial Change Visualization

Spatial change visualization allows the user to grasp *where* changes frequently occur in the video during a given period. This is done by presenting a spatial change visualization image (SCV image) that highlights regions where changes frequently occur during the period.

Figure 2 illustrates how to compute an SCV image. In this example, the video consists of four frames, and there are three frame differences. The SCV image is constructed by summing these three frame differences. Region A in the SCV image corresponds to the region in which changes occurred in frames 2, 3, and 4, whereas region B corresponds to the region in which changes occurred only in frames 3 and 4. Because region A has more changes than region B, it results in higher intensity in the SCV image.

In this example, we use only green to color the SCV image. However, we can also use red to enable spatial change visualization for two periods.

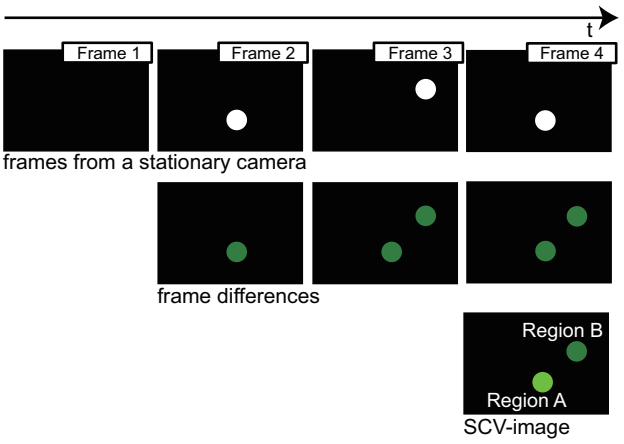

Fig. 2. An example of spatial change visualization.

B. Temporal Change Visualization

Temporal change visualization shows *when* changes frequently occur in a selected region during a given period. This allows the user to narrow in on the period that he or she needs to observe carefully.

C. Similarity-Based Video Retrieval

Similarity-based video retrieval allows the user to retrieve a portion of the entire video that is similar to the one in which he or she is particularly interested. For this purpose, we once more use frame differences.

To obtain the similarity of videos, we first introduce the similarity $\mathrm{sim}(a, b)$ between two images a and b (that are monochrome). We define it as the similarity between two lower resolution grayscale images obtained by reducing the resolutions of a and b but using a larger number of bits for each pixel. Specifically, we compute $\mathrm{sim}(a, b)$ as follows. First, from a and b, we construct their lower resolution images. Next we obtain the two vectors $\overrightarrow{a}$ and $\overrightarrow{b}$, whose elements are the pixels in the lower resolution images. Finally, we obtain $\mathrm{sim}(a, b)$ as the cosine of the angle θ between these two vectors:

$$\mathrm{sim}(a, b) = \cos(\theta) = \frac{\overrightarrow{a} \cdot \overrightarrow{b}}{|\overrightarrow{a}||\overrightarrow{b}|}.$$

We use the similarity of images to define the similarity of videos. Let A and B be videos that have the same number n of frames. Then we define the similarity $\mathrm{sim}(A, B)$ of A and B as follows:

$$\mathrm{sim}(A, B) = \frac{1}{n-1} \sum_{i=1}^{n-1} \mathrm{sim}(a_i, b_i),$$

where a_i and b_i for $i = 1, \ldots, n-1$ are the i-th frame differences obtained from A and B, respectively.

IV. Interface

Using the approach presented in the previous section, we implemented an interface for supporting the analysis of video taken with a stationary camera. This interface consists of a period window, a spatial pane, and a temporal pane. Figure 3 shows a screenshot of the interface.

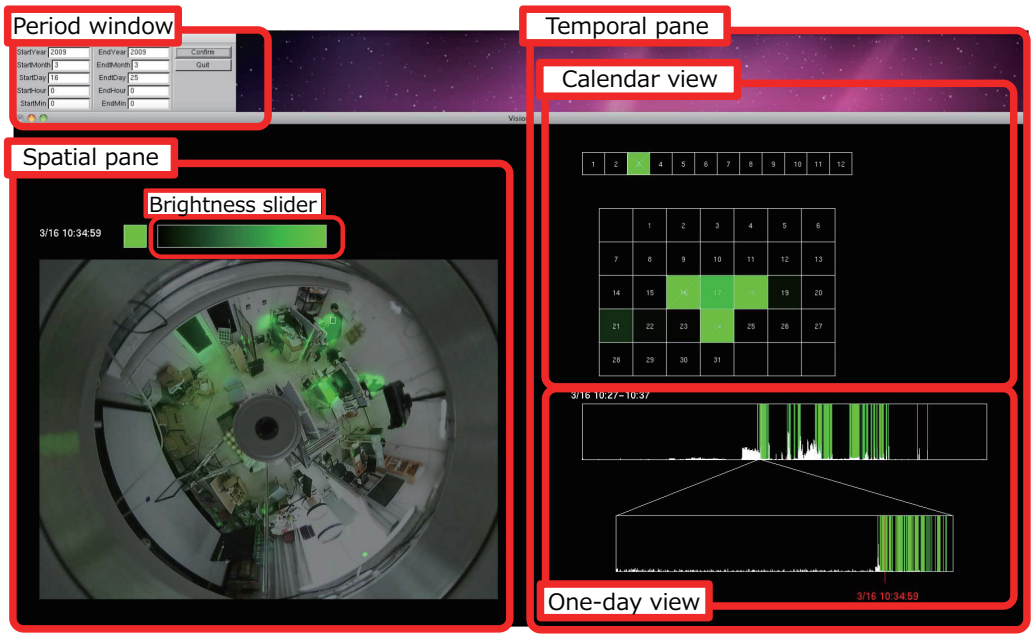

Fig. 3. Interface for supporting stationary camera video analysis.

A. Period Window

The period window provides text fields for specifying the beginning and end of the period for video analysis. To specify the period, the user enters the dates and times and presses the confirm button. Then the SCV image for the specified period appears in the spatial pane.

The interface allows the user to observe two periods at the same time. For this purpose, a second period window appears and lets the user to specify the second period.

B. Spatial Pane

For a period in the period window, the spatial pane displays the computed SCV image by superimposing it on the stationary camera image. It uses the stationary camera image captured at the beginning of the given period (although it initially uses a default image with no persons in the room). In our current implementation, it uses green and/or red to display SCV images. The SCV image is computed as described in Subsection III-A.

The spatial pane provides a slider for adjusting the brightness of the SCV image. The user can change the brightness value by dragging the slider. The image becomes brighter when the user drags to the right, and vice versa.

The actual brightness of the SCV image is computed by multiplying the original intensity of each pixel by the brightness value. Therefore, if the brightness value is small, only pixels with high intensity are visible, and thus the user can find regions in which many changes occurred. Furthermore, by gradually increasing the brightness value, the user can make other colored regions with less changes visible one after another. In this way, the user can identify the amount of change in various regions during the period by increasing and decreasing the brightness value.

The spatial pane is also used to select a region for temporal change visualization. Dragging the mouse over the displayed SCV image, the user can select a rectangular region for temporal change visualization. When this operation occurs, the temporal pane displays the results of the temporal change visualization focused on the selected region.

C. Temporal Pane

The temporal pane consists of a calendar view and a one-day view. The calendar view provides an overview and trend of the long-term temporal change visualization. The one-day view lets the user see the points in time when changes occurred in the selected region and also choose the time to view and the video used for similarity-based video retrieval.

1) Calendar View: The upper area of the calendar view provides the numbers 1 to 12, indicating months. The background color of each number shows the amount of change that occurred in the region (that was selected in the spatial pane) in the corresponding month during the given period. A brighter color means a larger amount of change.

By clicking on the number corresponding to a month, the user can view the days in the selected month in the lower area of the calendar view. As with the background colors of the months, the background color of each day shows the amount of change that occurred on that day. By clicking on a day, the user can obtain the one-day view for the selected day.

2) One-Day View: The one-day view consists of two areas, an upper and lower area, that provide temporal change visualization; the upper area is for a given day, and the lower area is for a 10-minute period selected by clicking on the upper area.

The horizontal axis in the upper area of the view indicates the time, with the left and right ends corresponding to 0:00 and 24:00, respectively, for the selected day. The one-pixel width in the axis corresponds to 2 minutes, and the time axis displays consecutive line segments, each of which is set to the color representing the amount of change during the 2 minutes. When the user clicks on some point of the axis, the temporal change visualization for the 10 minutes selected appears on the lower area of the view; the middle corresponds to the point clicked.

The horizontal axis in the lower area of the view also indicates the time, and the one-pixel width corresponds to 1 second. As with the upper area, the lower area displays consecutive line segments, each of which represents the amount of change for the corresponding time. Clicking on part of the lower area results in the spatial pane displaying the stationary camera image for the selected time.

In addition, the user can perform similarity-based video retrieval by using the one-day view. By dragging some part over the one-day view, the user can retrieve videos that are similar to the video corresponding to the selected period. This video retrieval is performed for the same day. The result is shown as white histograms in both the upper and lower areas of the view; a larger value in the histogram indicates a higher similarity.

V. Implementation

We are running a daemon that archives the video frames captured from an omnidirectional camera (Sharp Semiconductor LZ0P3551) mounted to the ceiling of our laboratory (the size is approximately 7.4 $\times$ 7.2 m) with a 816 $\times$ 608-pixels spatial resolution at 1 frame per second (fps), which is a frequently used frame rate in the video archives of surveillance systems. This frame rate leads the daemon to produce 86400 frames per day (77.8 GB in PNG on average in our case).

Therefore, key implementation issues for the efficient exploratory analysis of a long-term video include the following:

A. The tool should quickly show an SCV image in the spatial pane after the user selects the period(s) for analysis in the period window. Furthermore, the tool should quickly update the temporal pane after the user selects a region of interest by dragging over the spatial pane.

B. The faster the similarity-based video retrieval is, the higher the probability that the user will find potential events, both by counting similar events and by observing distributions of similar events in the long-term video.

The following sections describe how we implemented the tool to address these issues.

A. Optimizing the calculation of SCV images

The SCV image of a certain period is a blended image composed by adding all of the frame differences over the period. Because one frame difference is generated per second, composing the SCV image of a period $[s, s+n]$ in second requires adding $n+1$ frame differences in naïve implementation.

Fig. 4. Generating down-sampled frame differences for fast similarity-based video retrieval. Left: the original frame from a stationary camera with cut lines. Center: the corresponding frame difference with cut lines. Right: the down-sampled frame difference.

We optimized this process by pre-computing *accumulated frame differences*. Let $D_t(x, y)$ be the value of the pixel (x, y) of the frame difference at time t. Then $I_t(x, y)$, the value of the pixel (x, y) of the accumulated frame difference at time t, is defined as follows:

$$I_t(x, y) = \sum_{k=1}^{t} D_k(x, y).$$

Note that $D_1(x, y)$ is the frame difference computed when the daemon started. In our implementation, the daemon generates one accumulated frame difference per minute. The depth of each accumulated frame difference is 24 bits.

Using pre-computed accumulated frame differences, the computation of the SCV image of an arbitrary period requires only one subtraction. Let $D_{[t_1,t_2]}(x, y)$ be the value of the pixel (x, y) of the SCV image of the period $[t_1, t_2]$ $(t_1 < t_2)$. The SCV image of the period $[s, s+n]$ is

$$D_{[s,s+n]}(x, y) = I_{s+n}(x, y) - I_{s-1}(x, y).$$

By this optimization, the tool only has to load two accumulated frame differences to show an SCV image in the spatial pane after the user selects the period(s) for analysis in the period window. This results in the image being displayed very quickly.

Moreover, when the user selects a region of interest by dragging over the spatial pane, the tool uses accumulated frame differences to update the temporal pane. For example, to update the view of March 2011, the tool first loads the accumulated frame difference of 00:00 on March 1, 2011, and the one of 00:00 on April 1, 2011. Next it subtracts the former from the latter. Then, the tool sums the brightness of those pixels corresponding to the dragged region for the update.

B. Optimizing similarity-based video retrieval

The daemon also pre-computes lower resolution frame differences for similarity-based video retrieval. When the daemon generates a frame difference, it also cuts the left and right regions of the frame difference (see Figure 4 [left] and [center]), which can be considered to have a low possibility to contribute to detect events. The result is a 608 $\times$ 608-pixel image with 1-bit depth. Then the daemon down-samples it into a 19 $\times$ 19-pixel grayscale image with 8-bit depth. Figure 4 (right) shows the result. Although its resolution is very low, it still shows the regions in which a person moves.

In our implementation, the down-sampled frame differences are assembled into one file per day, both to minimize the

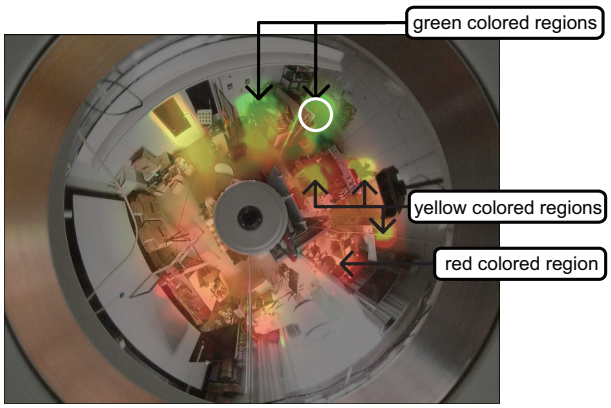

Fig. 5. Colored regions in an SCV image.

Fig. 6. The upper area of the temporal pane after the region around the white circle in Figure 5 is dragged.

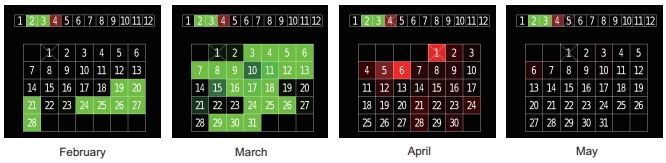

Fig. 7. Some calendar views in the temporal pane. These views were obtained by clicking the corresponding month in the upper area of the temporal pane.

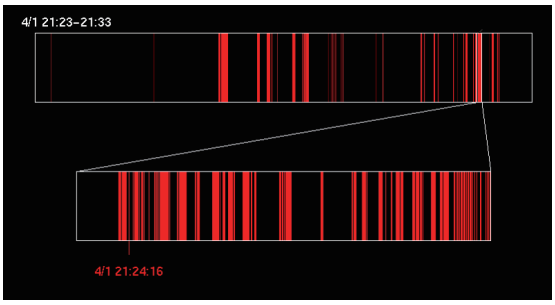

Fig. 8. One-day view of April 1. This view was obtained by clicking the corresponding day in Figure 7.

consumption of disk space and to reduce the time needed to load the images (31.2 MB in our case).

VI. Case Study

This section presents a case study to demonstrate how the three key components can be used to detect events and to examine findings in detail. In this study, we used a video archive recorded with a stationary camera (Figure 1) from February 19, 2011, to May 10, 2011 (81 days).

First we decided to examine whether any long-term changes occurred during this period. To this end, we divided the period into two halves: we set February 19 and April 1 as the start and end dates of the first half, respectively, and assigned green to this period using the period window; similarly, we set April 1 and May 11 as the start and end dates of the latter half, respectively, and assigned red to this period. The result was Figure 5.

This figure shows different-colored regions, especially desks used by students in our laboratory: two desks are green, three desks are yellow, and one desk is red. (Note that the combination of green and red yields yellow with the additive color method.) Therefore, this SCV image implies that two desks (green) were used in the first half but were empty in the latter half for some reason, three desks (yellow) were used during both periods, and one desk (red) became occupied in the latter half.

Next we decided to prove this inference by further examining one of the green regions, which is annotated with a white circle in Figure 5. By dragging the region on the spatial pane, we obtained Figure 6 on the upper area of the calendar view. The result was consistent with the above inference. In this figure, February and March are colored green with high brightness, whereas April and May are less bright. That is, there was more movement in the region before April than after April.

It is possible to further examine each month in detail by selecting a month in the upper area of the calendar view. The views in Figure 7 show the calendar view obtained by clicking February, March, April, and May, respectively, on Figure 6. These calendars also show that there was much movement in the region in February and March but little movement in April and May, except at the beginning of April (e.g., April 1 and April 6, which are rendered a very bright red).

Observing frames from the stationary camera would prove the inference. To this end, we obtained a one-day view of April 1 (Figure 8) by clicking the corresponding day in Figure 7. In this one-day view, the colored segments indicate the points in time when some movement occurred in the region. By mouse dragging these segments, the user can easily observe the corresponding frames. In this case study, we found several frames (e.g., frame 21:24:16 on April 1 in Figure 9) in which the student using the desk was packing various items, such as his computers and his belongings, in preparation for moving to another desk.

In summary, we were able to use our tool to detect certain events (i.e., red desks) and to quickly determine the reason why such events occurred. This was done by narrowing in on possible spatiotemporal regions and locating the frame sets that served as proof of our inference.

VII. Discussion

In the case study described in Section VI, we have found it greatly contributes to the user being able to detect interesting events that the system updates the SCV image instantly after the user selects a period of analysis whether the selected period of analysis is long or not. This enables the user to explore various period of analysis interactively, thus increasing the likelihood of the user detecting interesting events. At the same time, however, we also have found that the current implementation of the period window allows the user to only select

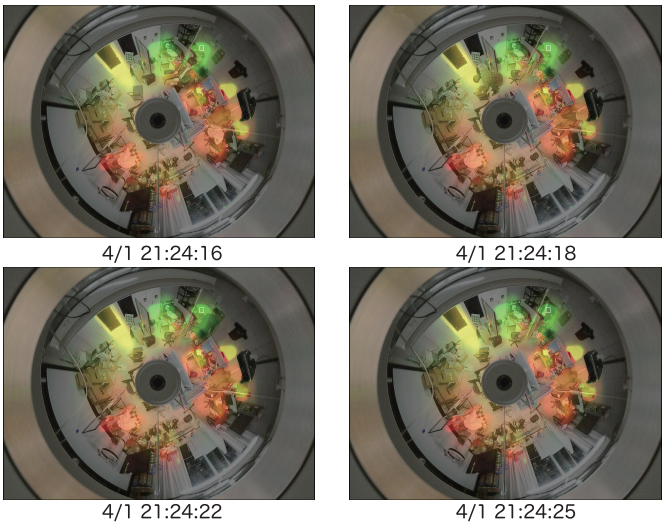

Fig. 9. Frames found in this case study.

consecutive periods of analysis, thus limiting possibilities for analysis. Therefore, we plan to improve the implementation to allow the user to select a period of analysis along with patterns, such as morning, a day of the week (e.g., Saturday), weekdays, and holidays.

Currently, the temporal pane shows the amount of change in different granularities from one year (the upper area of the calendar view) to one second (the lower area of the one-day view), simultaneously. We have found that this provides the user with a kind of focus+context visualization [18]. That is, while the user analyzes the amount of change in detail (i.e., using the lower area of the one-day view), he or she can always see what part of the period of analysis is being examining. This is because the temporal pane always shows the selected month, the selected day of the month, the selected period of the day, and the selected second of the period. However, the temporal pane cannot display the amount of change over one year. To address this issue, we plan to add a zooming feature, similar to [19], to the upper area of the calendar view, maintaining the focus+context visualization.

The user can perform similarity-based video retrieval within a day in the current implementation. Although we want to address this limitation, the cost of calculating similarity is proportional to the length of the period of analysis. Therefore, we plan to improve the calculation of similarity by making the implementation of the calculation parallel.

VIII. Conclusions and Future Work

We have presented a tool for the exploratory analysis of a long-term video from a stationary camera. This tool uses three key methods of analysis: spatial change visualization, temporal change visualization, and similarity-based video retrieval. These methods are uniformly realized in the sense that all of them adopt frame differences with 1-bit depth. We also have presented implementation techniques for optimizing the performance of these three methods.

Future work includes improving our tool as discussed in Section VII and conducting user studies that involve a longer video archive.

Acknowledgements

This work was supported in part by the FY2012 Joint Research Grant of the National Institute of Informatics, Japan.

References

[1] NEC Corporation, "Fieldanalyst," http://www.nec.co.jp/solution/video/fieldanalyst.

[2] Vitracom, "Siteview," http://www.vitracom.de/en/products.html.

[3] Y. Xing, Z. Wang, and W. Qiang, "Face tracking based advertisement effect evaluation," in *Proceedings of the 2nd International Congress on the Image and Signal Processing (CISP'09)*, Oct. 2009.

[4] A. Leykin, "Visual human tracking and group activity analysis: A video mining system for retail marketing," Ph.D. dissertation, Department of Computer Science and Cognitive Science, Indiana University, Dec. 2007.

[5] S. Fels and K. Mase, "Interactive video cubism," in *Proceedings of the 1999 Workshop on new Paradigms in Information Visualization and Manipulation (NPIVM '99) in Conjunction with the Eighth ACM Internation Conference on Information and Knowledge Management (CIKM '99)*, Nov. 1999, pp. 78–82.

[6] S. Fels, E. Lee, and K. Mase, "Techniques for interactive video cubism (poster session)," in *Proceedings of the eighth ACM international conference on Multimedia (MULTIMEDIA '00)*, Nov. 2000, pp. 368–370.

[7] A. W. Klein, P.-P. J. Sloan, A. Finkelstein, and M. F. Cohen, "Stylized video cubes," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Jul. 2002, pp. 15–22.

[8] E. P. Bennett and L. McMillan, "Proscenium: a framework for spatio-temporal video editing," in *Proceedings of the eleventh ACM international conference on Multimedia (MULTIMEDIA '03)*, Nov. 2003, pp. 177–184.

[9] T.-J. Hsieh, C.-K. Chen, and K.-L. Ma, "Visualizing field-measured seismic data," in *Proceedings of the 3rd IEEE Pacific Visualization Symposium (PacificVis 2010)*, Mar. 2010, pp. 65–72.

[10] R. P. Botchen, S. Bachthaler, F. Schick, M. Chen, G. Mori, D. Weiskopf, and T. Ertl, "Action-based multifield video visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 4, pp. 885–899, July/August 2008.

[11] M. Höferlin, E. Grundy, R. Borgo, D. Weiskopf, M. Chen, I. W. Griffiths, and W. Griffiths, "Video visualization for snooker skill training," *Computer Graphics Forum*, vol. 29, no. 3, pp. 1053–1062, Jun. 2010.

[12] S. Hashimoto and Y. Nakanishi, "SpaceTracer: Sharing space by compositing images from network cameras," in *Proceedings of Interaction 2006*. Information Processing Society of Japan, Mar. 2006, pp. 181–182, (in Japanese).

[13] G. Daniel and M. Chen, "Video visualization," in *Proceedings of the 14th IEEE Visualization Conference (VIS'03)*, Oct. 2003, pp. 409–461.

[14] M. Romero, J. Summet, J. Stasko, and G. Abowd, "Viz-a-vis: Toward visualizing video through computer vision," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, pp. 1261–1268, November/December 2008.

[15] C. Nguyen, Y. Niu, and F. Liu, "Video summagator: an interface for video summarization and navigation," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, May 2012, pp. 647–650.

[16] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson, "A system for video surveillance and monitoring," Carnegie Mellon University, Tech. Rep. CMU-RI-TR-00-12, 2000.

[17] D. A. Migliore, M. Matteucci, and M. Naccari, "A revaluation of frame difference in fast and robust motion detection," in *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, Oct. 2006, pp. 215–218.

[18] G. W. Furnas, "Generalized fisheye views," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, Apr. 1986, pp. 16–23.

[19] B. B. Bederson, "Fisheye menus," in *Proceedings of the 13th annual ACM symposium on User interface software and technology*, Nov. 2000, pp. 217–225.