Interactive Panning and Zooming the Virtual Camera

Stephan Fischer, Ivica Rimac, and Ralf Steinmetz
Industrial Process and System Communications
Department of Electrical Engineering and Information Technology
Darmstadt University of Technology
Merckstr. 25 • D-64283 Darmstadt • Germany
E-mail: {Stephan.Fischer, Ivica.Rimac, Ralf.Steinmetz}@kom.tu-darmstadt.de

Abstract

In this paper we explain how camera pans and zooms in digital video can be used to create a panoramic view of a video clip reordering the frames. As the view will normally be too large to be displayed on a single screen we propose the use of a grid where the user can move a "virtual camera" in different directions which are defined by camera pans as well as by camera zooms. To achieve this goal we explain in detail the architecture of our virtual camera.

Keywords: Video content processing.

1. Introduction

In this paper we describe a system to create a panoramic view of a scene of a video by analyzing the camera panning as well as zoom operations in that part of the video. The principle idea is as follows: if a stationary camera pans from left to right, then up and finally from the right to the left again a user would be able to move a "virtual" camera upwards directly by omitting the video frames which passed during the panning operation. This idea can be used to structure the content of surveillance videos. When looking for a burglar or some other event in a room being observed and recorded by a camera the police normally have to play a video tape until the interesting event occurs. This implies further actions to be performed, for example controlling the VCR by hand, rewinding and fast forwarding the video until the event is located. Using our system we compute a grid representing possible movements of the camera by which the temporal order of the video frames is changed. If a room has been recorded in detail arbitrary movements of the virtual camera become possible. An extension of our system recognizes zoom operations and uses these to construct a set of layered 2D-grids each of them offering a different resolution of the video together with the control operation of the virtual camera [4]. This allows for zoom operations completely independent from the original order of the images of a video.

The paper is structured as follows: Following a review of related work in section 2. Section 3 presents our approach to structure surveillance video. In section 4 we present experimental results we obtained using our system. Section 5 concludes the paper and gives an outlook.

2. Related Work

Work to create a panoramic view consisting of multiple video frames has been described by [8]. Mann uses single frames of a camera panning to compose a panoramic view of a video clip. The difference with regard to this paper is that Mann composes an image of greater size containing the visual representation of the camera panning while we show each frame of the sequence at a time and allow to control the direction of the camera movement reordering the video frame sequence.

Hypervideo approaches have been described by [12]. The approach differs from our approach as it is our goal to visualize the scene structure and hence to derive a control grid. In [12] the structure is used to create a hypervideo in space and time.

3. Controlling the Virtual Camera

Until now a user playing a video clip uses single frames together with the information of their temporal order. He can thus apply operations like play, fast forward or rewind at different speeds. These possibilities have been extended by algorithms to structure a video. An example therefore is the automatic detection of cuts in a video allowing a user to omit a scene and jump forward or backward to the next or

This work is supported in part by Volkswagen-Stiftung, Hannover, Germany.

to the last scene. However, the user still has to watch the video in the temporal order in which it was recorded.

Algorithms to calculate camera panning and zooming occurring in a video scene have been described in literature [6, 9, 10, 13]. We use our own approach to detect zooms [3, 4]. A scene is in this context defined as a sequence of single frames which does not contain any cuts or transition effects (wipes, fades or dissolves). Utilizing the calculated camera panning it is possible to create a tool with which a user can specify the direction of a camera pan from each frame of the sequence where images to the left and right as well as above and below are available. This can be achieved by analyzing the camera panning and by reordering the video sequence to create a movement grid for the user. If a camera first pans to the right, then up and finally to the left a grid can be calculated enabling the user to move the camera up directly. An example is shown in Figure 1.

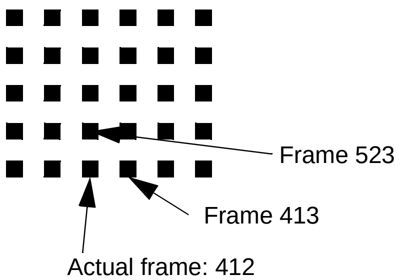


Figure 1. 2D camera movement grid.

A problem which results from the various combination possibilities of the images is caused by object motion. If a user omits too many frames in time when choosing a direction which is only possible due to calculations and not due to the real sequence of the video in time objects occurring in an image *i* can disappear in an image *(i+j)* and new ones can appear in an image *(i+j)* distracting the user. Although this happens in surveillance video quite rarely it should therefore be guaranteed that a movement in the camera grid is only allowed if no significant difference between two images in the camera grid can be found. A difference can very easily be computed for instance counting the number of pixels which changed from frame *i* to frame *(i+j)* in only that part of the images which can be found in both images (elimination of panning).

Once a two-dimensional grid representing the camera panning has been computed it can be extended to a three-dimensional grid by integrating zooms. 3D does in this context not refer to depth information but indicates that a set of 2D-grids is used between which the user can switch. In an ideal situation a room could be transformed into such a 3D-grid if and only if the room has been recorded by a camera in different zoom resolutions everywhere. Figure 2 shows an example of a 3D-grid.

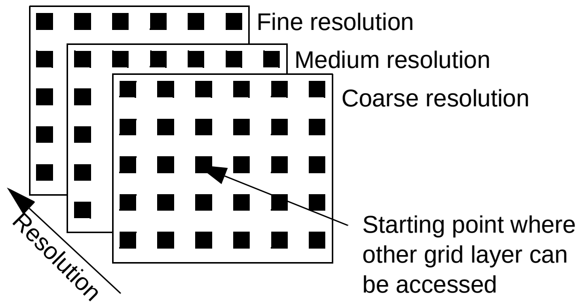


Figure 2. 3D camera movement and zoom grid.

The following applications can be created using this approach:

• Video surveillance systems. Rooms can be observed offline in an arbitrary direction if the respective images exist in the video material.

• Film producers can record a scene in different spatial detail and decide in the post production which level of detail to integrate into the final version of a film.

In the following we report on the experiences we obtained when developing the system.

3.1. Creation of a 2D-Grid

The calculation of camera panning can be used to derive a two-dimensional camera movement grid if no zoom operations are present in the video sequence being examined. It is obvious that errors during the calculation accumulate quickly in such a grid. The selection of a robust and precise algorithm must therefore be done very carefully. We chose the algorithm based on the Hausdorff distance to compute the camera panning as it fulfills the requirements of such a system at its best [9]. It should be noted that a computation of the Hausdorff distance in real-time is in most cases impossible. In the case of surveillance video this does not matter as the video can be precomputed. The result of the computation of the Hausdorff distance between two images (I_i, I_{i+1}) is a vector *(u, v)* which corresponds to the camera panning between those two images.

Using the formula

$$\sin(A) = \frac{v}{\sqrt{u^2 + v^2}},$$

the angle *A* of the movement can be calculated, using the formula

$$L = \sqrt{u^2 + v^2}$$

the length *L* of the movement can be calculated.

Having calculated the angle *A* and the length *L* of the movement a control system using 8 different movement directions can be created. 8 directions are necessary to allow the user to move the camera horizontally, vertically and also diagonally. The control interface is shown in Figure 3, a grid in Figure 4. Note that the control interface does not permit to move the camera to the left and to the lower right in this example.

This is due to the fact that our recording did not cover the whole outdoor scenery and that certain movements are not allowed (see below).

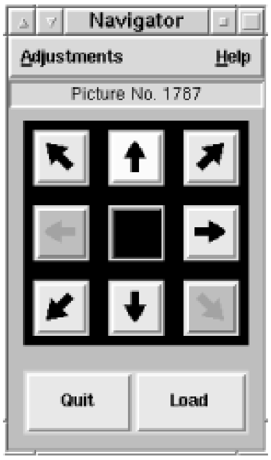
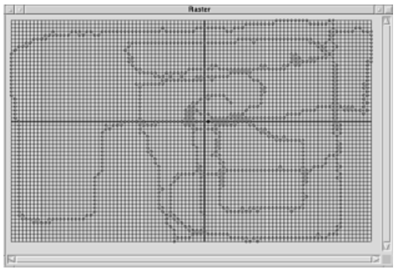


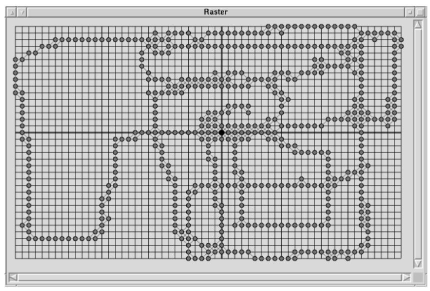
Figure 3. 2D control interface.

As the camera is allowed to pan in an arbitrary direction a first step of the calculation is to represent the vector chains in a continuous plane. To be able to display a grid the continuous plane is then quantized following the rule that a position of the continuous plane is assigned to that of 8 neighbors of the discrete 2D-grid to which the distance is minimal. As it is possible that multiple frames exist in the neighborhood of a discrete point of the grid the one nearest to the grid has to be chosen. The grid represented in Figure 4 has been calculated using a short video clip which has been recorded at our institute using a camera and a tripod. The panning which took place is clearly visible. In addition Figure 4 shows two different resolutions of the grid. A user who is interested in a fast pan of the virtual camera can choose a different resolution of the grid which has been calculated using a different quantization coarseness of the continuous vector plane. This functionality is hence comparable to some kind of fast forward in an old-fashioned video. The center of both grids which is also indicated by two crossing thick lines represents the actual frame the user is watching.

(a) Image of video sequence



(b) Fine resolution grid



(c) Coarse resolution grid

Figure 4. Outdoor scene and corresponding grids.

3.2. Creation of a 3D-Grid

As already explained above the grid structure can be extended to a layered structure using a set of different grids to represent different zoom resolutions of a video sequence. If such a sequence contains for example a zoom-in followed by panning operations and a zoom-out of inverse zoom factor two grids can be used where the begin and the end of the respective zoom operation can be used to switch between two grids with different resolution of the video.

Zoom factors can be calculated as described in [3, 4]. A grid with another resolution can then be calculated using the zoom to attach the second grid to the first. As zoom factors don't have to be whole-numbers the same quantization as in the case of the creation of the two-dimensional grid has to be applied, this time using a third dimension (resolution dimension). Quantization results are provided in section 4..

3.3. Definition of movement within a Grid

Having computed the different grids a three-dimensional structure representing the video frames and the possible movement is available. The number of horizontal or vertical elements of each of the two-dimensional grids equals at most the number of frames a video sequence contains. This can only happen if the sequence consists of only one horizontal or vertical camera pan during the whole sequence. If the application of the system is not constrained to surveillance video it must be defined if the movement from a point P_{ijk} with coordinates i and j of a certain grid resolution k to one of the neighboring points can be permitted as the elimination of the temporal order of the frames of the sequence can lead to objects which appear or disappear all of a sudden from one frame to the next thus confusing the user. In the case of surveillance video this normally does only happen if an event occurs the user is interested in anyway. In this context the term *allowed* is equivalent to a certain similarity of two frames which must be significant. The following problems can arise prohibiting a movement from a frame A to a neighboring frame B:

- the brightness of the images changed significantly during a zoom or a pan operation.
- although close together in the grid the images are different due to object movement.

Various approaches to analyze the similarity of images have been described in literature. We analyzed difference images as well as the Hausdorff distance measure. Difference images can be calculated quickly either by subtracting the images or by comparing their histograms. Both approaches require the compensation of camera panning before comparing the images. The use of histograms is not very advantageous as object movement is eliminated when working with a single histogram for an image. A second disadvantage of difference techniques is their strong reaction to changes of brightness. Even if two images contain the same content they will appear very different if the brightness changed. The Hausdorff distance measure is much more robust but needs considerably more computation time. Experiments showed that the computation of the Hausdorff measure is app. 50 times slower than that of difference images. As it becomes quite complex to precompute the image similarities particularly as different quantization factors shall be allowed we used pixel difference images to analyze the image similarity. If the similarity of a point P_{ijk} with coordinates i and j of a certain grid resolution k to one of the neighboring points is significantly high a movement between those two images is allowed. Experiments showed that a similarity above 90 percent is a good value to obtain satisfactory results. This effect can also be observed in Figure 3. While a movement to the left is not allowed because of missing video material in that direction the movement to the lower right is prohibited because of a low image similarity. We compared both images manually and found that a tram showed up in the frame to the lower right (object movement) which lowered the image similarity. The control interface in Figure 3 hence has different colors for movement which is allowed and for directions which are prohibited.

We also use the similarity measure to calculate different resolutions of a single grid. If the comparison is no longer based on neighbors (radius 1) but instead based on images which are in a distance of two images in the grid (radius 2) the resolution of the grid becomes coarser enabling the user to parse the video faster. In Figure 4 an original image and two grids of different resolution representing the same video sequence are shown.

4. Experiments

We implemented our system using a combination of Tcl/Tk and C++ where Tcl/Tk was used to develop the interfaces and the grid representation and C++ was used to calculate the pans and zooms as well as the image differences. Experiments showed that an image similarity of at least 90 percent as well as a Hausdorff similarity of at least 40 percent between two images neighbored in the grid are sufficient to allow the movement of the virtual camera without artifacts in the movement of the virtual camera.

The sequences which were used in our experiments could be transformed into a set of grids without problems. A disadvantage we did not expect was that we recorded the outdoor scenes without great care. The resulting grid contained regions where the user could arbitrarily move the virtual camera but also motion trails where no neighboring images were available. As a logical consequence the motion trails without neighbors correspond directly to the temporal order of the video sequence. Normally rooms being observed by a surveillance system are rather small. A recording where many overlapping areas occur is therefore to be expected. Considering the motion trails without neighbors in the grid, the structure of the video could be visualized offering the possibility to play exactly those parts of the video which contained portions of the outdoor sequence which are of particular interest.

5. Conclusions and Outlook

In this paper we propose a new method for reordering the images of a video sequence in order to create a panorama view of that sequence. The temporal order of the respective images is thus transformed into a “spatial order”. The user is hence enabled to move a virtual camera within a sequence to an arbitrary direction if recorded by the camera and if no significant object movement prohibits a move between two adjacent points of the spatial grid. A condition for that work is the use of a stationary camera and the constraint that object movement does not play an important role in the content of the video sequence.

In order to create a set of grids representing different resolutions we described how camera panning and the recognition of zooms have to be used to create the virtual camera system. As no algorithms are known yet to recognize a zoom precisely we propose a new method to recognize zooms and to calculate a precise scaling factor between successive frames of a video sequence.

An interesting issue is how to handle multiple frames available at a specific grid position. This happens very often as the camera passes the same position over and over again. If nothing changed the image nearest to the respective grid position has to be displayed thus avoiding redundant frames. Redundancy can be measured efficiently comparing the similarity of frames being available for the same grid position. Also the case of images of different content at the same grid position has to be considered indicating that an event happened the user might be interested in. We currently examine if a panel can be used which can be popped up containing the different images for the same grid position.

The system we described in this paper only uses a stationary camera. We currently extend the system to use multiple cameras to enable the user to change the camera position. Imagine a video sequence where the user can choose a specific position where to be located within a scene. We thus examine how the images of different cameras have to be combined and how different resolutions available can be used to zoom not only to an arbitrary resolution but also to change the direction of sight thus watching what happens during the temporal order of a video.

Acknowledgment

We would like to thank Wolfgang Effelsberg from the University of Mannheim and his group. This work is a result of many discussions and advice given by the MoCA group.

References

[1] J. S. Boreczky and L. A. Rowe. *A comparison of video shot boundary detection techniques*. Journal of Electronic Imaging, 5(2):pp. 122-128, 1996.

[2] M.J. Buckley. *Fast computation of a discretized thin-plate smoothing spline for image data*. Biometrika, 81(2), pp. 247-258, 1994.

[3] S. Fischer. *Feature combination for content-based analysis of digital film*. PhD thesis, University of Mannheim, 1997.

[4] S. Fischer. Automatic Recognition of Camera Zooms. To appear in Proc. of ViSual, Amsterdam, 1999.

[5] M. Hötter. *Differential Estimation of the Global Motion Parameters Zoom and Pan*. Signal Processing, 16, pp. 249-265, 1989.

[6] K. Illgner, C. Stiller, and F. Müller. *A Robust Zoom and Pan Estimation Technique*. Proceedings of the International Picture Coding Symposium PCS’93, Lausanne, Switzerland, 1993.

[7] B. Lucas and T. Kanade. *An iterative image registration technique with an application to stereo vision*. DARPA IU Workshop, pp. 121-130, 1981.

[8] S. Mann. *Smart Clothing: Wearable Multimedia Computing and Personal Imaging to restore the technological balance between people and their environments*. Proc. ACM MM 1996, pp. 163-174, Boston, 1996.

[9] K. Mai, J. Miller, and R. Zabih. *A Feature-based Algorithm for Detecting and Classifying Scene Breaks*. Proc. ACM MM 1995, pp. 189-200, San Francisco, 1995.

[10] M. Pollefeys, R. Koch, and L. Van Gool. Self-Calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters, Proc. of ICCV, 1998.

[11] W. Rucklidge. *Efficient computation of the minimum Hausdorff distance for visual recognition*. Dept. of computer science, Cornell University. TR-94-1454, September 1994.

[12] N. Sawhney, D. Balcom, and I. Smith. *Authoring and Navigating Video in Space and Time*. IEEE Multimedia Journal, Fall 1997.

[13] Y.T. Tse and R.L. Bakler. *Global zoom pan estimation and compensation for video compression*. Proceedings ICASSP, pp. 2725-2728, 1991.