Video Visual Analytics of Tracked Moving Objects

Markus HÖFERLIN [a],Benjamin HÖFERLIN [b] and Daniel WEISKOPF [a]
[a] *Visualization Research Center, Universität Stuttgart, Germany*
[b] *Intelligent Systems Group, Universität Stuttgart, Germany*

Abstract. Exploring video data by simply watching does not scale for large databases. Especially, this problem becomes obvious in the field of video surveillance. Motivated by a mini challenge of the contest of the IEEE Symposium on Visual Analytics Science and Technology 2009 (Detecting the encounter of persons in a provided video stream utilizing the techniques of visual analytics), we propose an approach for fast identification of relevant objects based on the properties of their trajectories. We present a novel visual and interactive filter process for fast video exploration that yields good results even with challenging video data. The video material includes changing illumination and was captured with low temporal resolution by a camera panning between different views.

Keywords. Visual analytics, video surveillance, video visualization

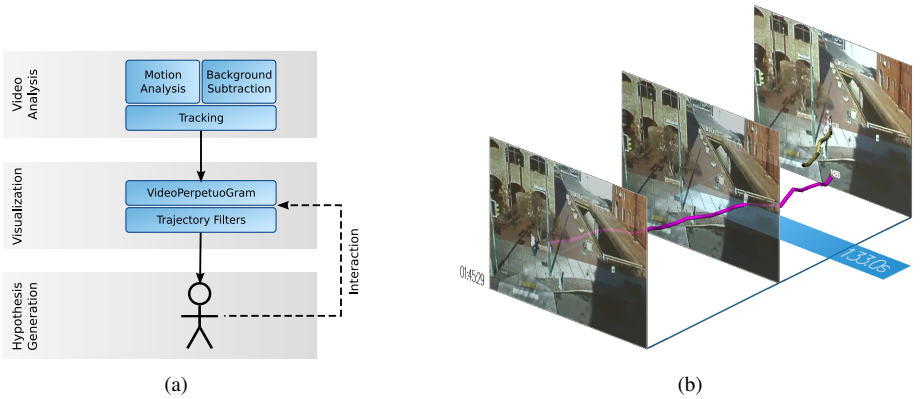
Introduction

Over the last few years the amount of CCTV cameras has been increasing rapidly, especially but not solely in the field of video surveillance. For example, the human rights group Liberty estimated about 4.5 million CCTVs for the UK in 2009 [1]. That is one CCTV camera for every 14th citizen. On the one hand, directly watching video footage does not scale up with the growing number of CCTVs. Even if we assume an operator to watch multiple video streams in fast-forward mode this ends up in a cost expensive process: Haering *et al.* [2] put the cost for monitoring 25 cameras by human observers to $150k per annum. Additionally, the attention of an operator decreases within 20 minutes [2]. On the other hand, fully automated computer vision systems that process video data to a high semantic level are not reliable yet [3].

A solution to these problems is provided by the visual analytics (*VA*) process, which is situated between the two mentioned extreme cases: fully automated video analysis and manual video analysis. Therefore, VA combines automated video analysis on lower semantic levels with an appropriate visualization. For the classification of these features, VA relies on human recognition abilities, linked to the system by interaction. This enables an accelerated exploration of video data relying on the specialized capabilities of each: human and computer.

Contributions: Based on the VA methodology, we propose a novel framework for video analysis and evaluate it, using the IEEE VAST Challenge 2009 video data set [4]. The

structure of the proposed VA framework is depicted in Fig.1(a). In a preprocessing step, we analyze the moving objects of a video and apply established approaches like optical flow computation, background subtraction, and object tracking by a Kalman filter. Further we visualize the video as a *VideoPerpetuoGram* (*VPG*) [5]. As second contribution, we enable the users to interact with the VPG by defining filters. Thus, parts of low interest can be neglected, while users can focus on relevant periods. By this interactive visualization and filtering process, users are enabled to refine their hypotheses in an iterative manner and thus, the VA process is completed.



(a) (b)

Figure 1. The structure of the proposed system is illustrated in (a). An example of the VPG is presented in (b). Three keyframes are displayed with their temporal positions aside. A blue bar represents the amount of time skipped. The pink and beige tubes indicate two trajectories.

Related work: The field of *VA* as described by Thomas *et al.* [6] is quite young and rapidly growing. It addresses the challenging and highly relevant task of analyzing huge amounts of complex data that cannot be processed fully automatically in a reliable manner.

Prior work on visual *video abstraction* condensed actions in a temporal matter by showing several actions at the same time even if they occur chronologically in succession [7]. Caspi *et al.* [8] selected informative poses of objects and merged them into images or short video clips. Their approach dealt with occlusion by rotating and translating the poses in the video volume.

Other works introduced *video browsing* techniques for better video exploration. For example, Dragicevic *et al.* [9] proposed to browse interactively through videos by direct object manipulation. This is the selection of a temporal video position by dragging an object of the video to the desired spatial position.

The foundations of the *VPG* were developed during the last years. The first who rendered a video as 3D volume were Fels and Mase [10]. Daniel and Chen [11] mapped the 3D volume to a horseshoe and additionally displayed image changes with the aim of video summarization. Chen *et al.* [12] deployed flow visualization techniques to video volumes and identified relevant visual signatures in video volumes. Finally, the VPG was introduced as a seismograph-like visualization technique for continuous video streams by Botchen *et al.* [5].

1. Video Vision

The part of video analysis depicted in Fig. 1(a) applies established computer vision techniques. We segment temporally changing regions combining background subtraction and optical flow computation. Both methods on their own are not reliable enough to analyze challenging data sets like the one provided with the VAST Challenge 2009, which includes changing illumination and was captured with low temporal resolution by a camera panning between different views. Thus, we use background subtraction and optical flow segmentation in a complementary manner. After segmenting the foreground blobs we associate them with the blobs detected in the previous frames utilizing a Kalman tracking filter. Finally, different properties of these trajectories are calculated, which are used in the subsequent steps.

Background Subtraction: Background subtraction is the specialized case of change detection primarily used for foreground segmentation of video sequences captured by static cameras. Although there exist sophisticated approaches of background models, we go for a very basic but robust method able to cope with the lack of training data. The model of each background (one for each camera position) is calculated as median of the last 150 frames. Violations to this background model are considered as foreground regions and are calculated as luminance distances between model and sensed image that are above a certain threshold. After the segmentation step we update the background model, using the most recent frame. Since the camera does sometimes change its viewing direction, a precise overlap of the actual viewing volume with the background model cannot be assumed. Therefore, we have to realign the sensed image with the model by translating the image to the maximal cross-correlation response within a region of several pixels from its initial position. The median is a convenient method to shape an adaptive background model because it is statistically insensitive to outliers originating from noisy video data. Due to the rotation of the camera, static changes (i.e. scene changes that are uncorrelated to any motion) affect the background subtraction. That happens if the scene changes while the camera points into another direction. This results in background violations of non-foreground objects.

Motion Segmentation: Another common concept to extract relevant objects of a video sequence assumes that these objects are moving. These objects can be identified by segmenting regions with homogeneous motion and a velocity above a certain threshold. For motion analysis we rely on the pyramidal Lucas-Kanade method [13]. Subsequently, we segment the motion field based on motion homogeneity. By applying motion segmentation we are able to reject regions originating from a badly initialized background model or static changes during camera rotation. Motion segmentation on its own would lack robustness in cases of strong video noise.

Tracking of Segmented Regions: By tracing the detected regions over several frames we build up their trajectories. These trajectories are the principal objects we use for further visual analysis. A linear Kalman filter up to the third order is used to track the detected region's position and size in image space.

Properties of the Trajectories: As final step of the video analysis, several properties of the extracted trajectories are calculated. Among them are the object's speed, average direction, and distance to other trajectories at its start and end positions. To obtain these information we homographically project the trajectory's position onto the top-view plane and measure the distance in world space.

2. Video Visualization

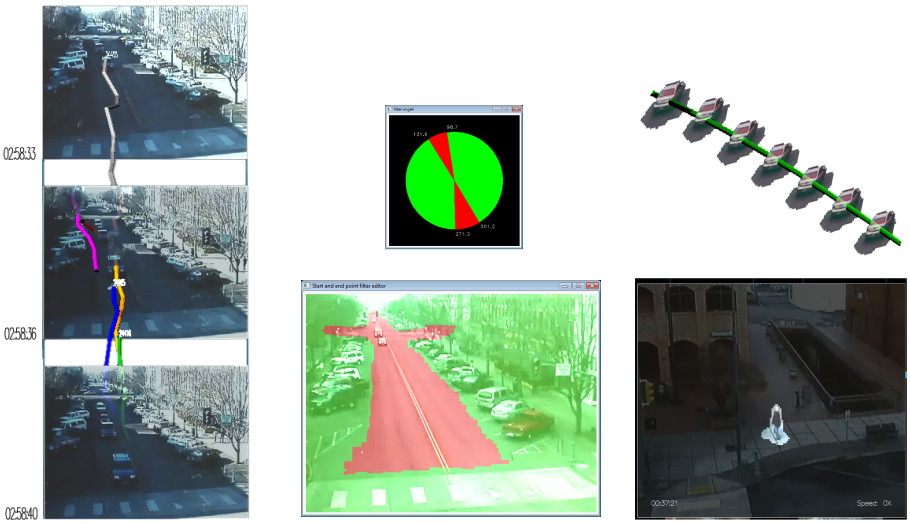
We propose a video visualization approach based on the VPG by Botchen *et al.* [5]. The VPG is a visualization technique that enables continuous video streams to be displayed in a manner similar to seismographs and electrocardiograms (ECG). The two spatial axes of the video are extended by time as third axis, yielding a 3D video volume. Inside the volume, keyframes are displayed at sparse, equidistant intervals to convey context information. Trajectories extracted in the preprocessing step are included in the volume and reveal movement information. The video sequence is split into independent camera views each represented by a VPG side by side. An example for one camera direction is illustrated in Fig. 1(b). Additional, blue bars inside the volume indicate skipped time intervals. Their durations are plotted onto the bars. Time is skipped between every scene change. If there is no relevant content within a period or a scene, we omit it. The relevance of a scene is defined by filters that we will discuss in the next section.

3. User Interaction

To cope with the large amount of video data, we enable the users to interact with the visualization by applying filters and thus achieve scalability. Trajectories are filtered by their relevance according to properties like camera location, temporal and spatial start and end positions in image coordinates, mean speed, and average direction. Complex filters can be created by defining arbitrary numbers of filters concatenated by logical operators like *AND* and *OR*. Beyond that, it is possible to apply trajectory interaction filters. These filters empower the users to focus on trajectories by specifying their relation to other trajectories. For example, an interaction filter may require that trajectories have to begin or end in spatial and temporal vicinity. The aim of filtering is to decrease the number of trajectories and to focus on regions of interest.

To gain a deeper understanding, we provide an example related to the VAST Challenge. In the VPG illustrated in Fig. 2(a) a lot of trajectories are extracted that originate from moving cars. Since we are searching for encounters of people, these trajectories are not relevant to us. Therefore, we add a spatial start and end position filter in image coordinates (cf. Fig. 2(b)(bottom)). Another possibility to reject the trajectories of cars is to filter the average direction as depicted in Fig. 2(b)(top). Since cars are typically faster than pedestrians, the application of a mean speed filter would be an option, too.

In VA, the interaction is typically an iterative process guiding the hypothesis generation. The users infer hypotheses by inductive and deductive steps based on the visualization. The typical scenario looks like this: First, the unfiltered video volume is observed. The users explore the video and identify some uninteresting events, e.g. pedestrians waiting at a pedestrian crossing. Thus, they define filters to ignore the trajectories according to their features. By further exploration they detect other events without relevance, e.g. pedestrians just crossing the scene or trajectories that do not affect each other. These will also be neglected, decreasing the number of trajectories in the result set. The filtering process does not necessarily narrow the set of trajectories, but can also widen them by unifying the result sets of two filters. Uninteresting events can be ignored using a black list. This way, the users iteratively build hypotheses based on their exploration of the video. Simultaneously, they verify their hypotheses by defining an appropriate set



(a) (b) (c)

Figure 2. (a) VPG with many trajectories of car movement. (b) Positional (bottom) and directional (top) filter definition. (c) Blobs along a trajectory, extracted by the preprocessing step (top). Highlighted moving object in the video sequence (bottom).

of filters. These steps help the users to reduce the amount of video data that remains for watching. Finally, the users obtain a manageable amount of data, small enough for a detailed manual analysis. Note that defining filters in this way will not lead to a semantic gap, since the formulation of filters and the visualization directly depend on low-level features.

To gain confidence on the automatic video analysis of the preprocessing step, the users are able to examine the foundations from which the visualized data is inferred. The volume slices of the trajectories (cf. Fig 2(c)(top)) or the playback of a part of the video sequence showing the traced object highlighted (cf. Fig 2(c)(bottom)), are tools to serve this purpose.

We point out the advantage of our approach over the recent method of watching the whole video sequence, using the task provided by the VAST Challenge 2009. The task was to find the encounter of people within 10 hours of video surveillance material. In contrast to manually inspecting the 10 hours of video footage, we begin the proposed VA process with an amount of 809 trajectories and the initial hypothesis provided by the VAST Challenge's task. After a short period of time an experienced user was able to reduce the number of remaining trajectories to 22. Similar as described in the examples above, this was achieved by an iterative refinement of the hypotheses and the applied set of filters. Especially, the application of an interaction filter adjusted to detect the split and merge of trajectories was able to condense the numbers of remaining object's traces. Finally, a suspicious encounter of two people could be tracked and validated by the user.

4. Conclusion and Future Directions

In this paper we have proposed a method for scalable video analysis based on the visual analytics methodology. Reliability issues arising with fully automated video analysis approaches are avoided by involving human recognition abilities. We have proposed a novel framework that consists of three building blocks providing scalability to large quantities of video data. First, a video sequence is automatically analyzed on a low semantic level. Extracted features are then visualized in relation to the original content using the VPG. As third and principle concept, the users interact with the system and apply filters based on iteratively refined hypotheses. Finally, we have illustrated the usefulness of our approach by an example derived from the VAST Challenge 2009.

Future work could consider other features than trajectories. Also, additional confidence information of these features and their relation to each other will increase reliability and scalability. An important area of future research is a more detailed evaluation by quantitative user studies.

Acknowledgments

This work was funded by DFG as part of the Priority Program "Scalable Visual Analytics" (SPP 1335).

References

[1] Liberty. Closed circuit television - CCTV. [Online]. Available: http://www.liberty-human-rights.org.uk/issues/3-privacy/32-cctv/index.shtml

[2] N. Haering, P. Venetianer, and A. Lipton, "The evolution of video surveillance: an overview," *Machine Vision and Applications*, vol. 19, no. 5, pp. 279–290, 2008.

[3] A. Dick and M. Brooks, "Issues in automated visual surveillance," *New Scientist*, pp. 195–204, 2003.

[4] IEEE VAST Challenge 2009. IEEE VAST 2009 Symposium. [Online]. Available: http://hcil.cs.umd.edu/localphp/hcil/vast/index.php

[5] 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, 2008.

[6] J. Thomas and K. Cook, *Illuminating the path: The research and development agenda for visual analytics*. IEEE Computer Society, 2005.

[7] Y. Pritch, A. Rav-Acha, and S. Peleg, "Nonchronological video synopsis and indexing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1971–1984, 2008.

[8] Y. Caspi, A. Axelrod, Y. Matsushita, and A. Gamliel, "Dynamic stills and clip trailers," *The Visual Computer*, vol. 22, no. 9, pp. 642–652, 2006.

[9] P. Dragicevic, G. Ramos, J. Bibliowitcz, D. Nowrouzezahrai, R. Balakrishnan, and K. Singh, "Video browsing by direct manipulation," in *Proceeding of the 26th Annual SIGCHI Conference on Human Factors in Computing Systems*. Florence, Italy: ACM, 2008, pp. 237–246.

[10] S. Fels and K. Mase, "Interactive video cubism," in *Proceedings of the 1999 Workshop on New Paradigms in Information Visualization and Manipulation (NPIVM)*. ACM NY, USA, 1999, pp. 78–82.

[11] G. Daniel and M. Chen, "Video visualization," in *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*. IEEE Computer Society Washington, DC, USA, 2003, pp. 409–416.

[12] M. Chen, R. Botchen, R. Hashim, D. Weiskopf, T. Ertl, and I. Thornton, "Visual signatures in video visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1093–1100, 2006.

[13] J. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker: description of the algorithm," *OpenCV Documentation, Intel Corp., Microprocessor Research Labs*, pp. 593–600, Jun 2000.