Analyzing video produced by a stationary surveillance camera

Paolo Buono
Dipartimento di Informatica
Università degli Studi di Bari "Aldo Moro"
via Orabona, 4, Bari, Italy
buono@di.uniba.it

Abstract — **Today surveillance systems are everywhere. Human observers watching live videos of specific areas are not efficient due to the likely loss of attention. On the other side, unattended surveillance systems require that people analyze hours of recordings when they have to search for some specific events, e.g. identify people responsible of violence, theft or other offences. In many cases a specific search in the video has to be accomplished in the shortest amount of time. This paper presents MotionFinder, a tool that performs video analysis by computing an interactive summarization of the movements in a scene. Once the summarization process is complete, the tool responds in real time to inquires. For example, human investigators may search for specific areas in the video that show high levels of activity or where they know that something occurred (e.g.: property damaged or stolen). The tool responds by showing only the scenes in which some activity occurred for that specific area of the video.**

Video summarization, video analysis, visual analytics, stationary surveillance cameras

1 Introduction

One morning, John enters into his work building and he notices that the director, the responsible of security, the guardian and a technical employee are looking at the guardian's monitor in the guardian's room. After a couple of hours, John passes by the guardian's room again and the four people are still there, so John asks what is going on. It took about four hours of the four people to get to the exact moment in the video in which a thief stole a laptop from an office. The four people moved back and forth in the video and also used the fast-forward feature without noticing any change in the office scene, since only a few frames within a video recording of about 24 hours referred to the very quick action of the thief getting the laptop from the office.

The previous scenario describe a typical situation in the video surveillance context. Today video surveillance stationary cameras have built-in software able to recognize movements in the scene, i.e. to capture scenes in which something moves. Motion detection doesn't always helps, in particular for heavy traffic areas, where almost always something moves.

Having a human guardian watching surveilled areas could be not optimal, particularly when there are many cameras. Human attention span can drop below acceptable levels after only 20 minutes, even in trained observers [4]. If we add that humans can handle a number of items of seven, plus or minus two, the use of observers is very ineffective when more than a tenth of video surveillance cameras are installed. Nevertheless, a surveilled area is a deterrent for people and reduces the number of causalities or damages. Entire cities have networks of surveillance cameras in order to cover specific locations for detecting and identifying potential threats or suspicious events. These systems often adopt real-time algorithms for detecting anomalies, identify objects and track them.

This paper describes a novel technique we have developed for video analysis and a sofware tool, called MotionFinder, which implements this technique. A first proposal of the technique was presented in [2]. MotionFinder allows a human investigator to speed up its search for anomalies by quickly selecting excerpts of the video in which an event occurred. The tool is intended for post-processing activities, not for providing real-time alerts. Nevertheless, it is possible to work in real-time, since the adopted summarization technique is very fast.

Next section provides an overview of related work. In Section 3, the summarization technique is presented. MotionFinder is illustrated in Section 4. Section 5 describes an interaction session in order to show how the tool works. Finally, conclusions and possible future research directions are reported.

2 Approaches in video surveillance

This section reports the most relevant work about the current video analysis systems. Most of the deveoped systems try to solve the problem of video analysis by creating fully automated applications. In video surveillance, events and requirements are often unpredictable, making useless the software used for the analysis. In effective video analysis it is important to combine the perception, flexibility, creativity and general knowledge of the human mind with the enormous storage capacity and computational power of computers [8]. Because analysing video is time-consuming, researchers try to to support the analysts by allowing them to focus their efforts on relevant parts of the video, avoiding wasting precious time on irrelevant segments.

A significant research has been done in building fully automated systems, which should leverage human investigators in analyzing the videos. A survey on the sate of the art about automated visual surveillance technologies is [13]. The survey focuses on intelligent surveillance systems, which use techniques and methods for recognizing objects and humans with the aim to describe their actions and their interactions. Techniques and tools are presented focusing on aspects like object recognition, behavioural analysis, and surveillance systems architectures. Instead of focusing on automated techniques, our goal is to provide human investigator with tools that empower their abilities.

In [5], beside a discussion about the state of the art for object recognition, particular attention is given to the architecture and the user interface, which should allow users to easily operate with the system. The authors build a "surveillance index browser", which is part of a larger architecture that includes a face-recognizer module. In the user interface, a timeline represents the overview of all the events detected by the system in a particular time-frame. A second timeline provides a zoomed version of any video segment chosen by the user. A window displays the output of the tracker camera, while a second window displays a zoomed-in video about the moving object. Moving objects can be shown as color coded traces; according to time, older events are drawn in white, which gradually turn in red as time passes. In order to limit screen cluttering, the user can also apply filters. Also our tool uses a timeline, a window for viewing the video, and the color coded traces.

Performing queries in video databases is another requirement of video analysis. The system of Huston et al allows the human investigator to perform queries on unstructured data in a large number of distributed camera located in a city area [6]. The goal is to take advantage of the computer resources for searching video data, allowing the user to focus the investigator attention on interpreting the results in the hope of gaining insights that might help to accomplish the task the user is undertaking. A query is triggered by selecting a portion of an image; the system starts the search using a brute-force algorithm, which is the only viable solution, because the user is free to select any region of any frame of the video, and generally it is not expected to know what to search for, making indexing useless. Moreover, the authors point out that the rate at which new data is generated far exceeds the rate at which it can be analysed, and most of it will never be searched before being erased; as a consequence, pre-processing can be seen as a waste of resources. The human investigator can select different search parameters, like color, shape, texture and object detection. In order to avoid long waiting, the system shows results as soon as they are found and allows users to perform concurrent searches. In the tool proposed in this paper, the analysis performed by the user is not known in advance, but indexing is performed because: 1) the produced image can be used as a visual search for videos, since it is considered as a preview of the video; 2) indexing process is faster than the production of the video itself; 3) index files are very small.

Some thoughts about video occupancy space are provided by Romero et al. [10]. They address a system that tracks over 7500 hours per month, which occupy about 7500 GB space (240 GB per day). This means that it is unfeasible to use a typical laptop for the analysis. In order to be able to keep in a laptop one month of recorded videos, they exclude all frames in which no activity is detected; in this way, the needed space for one month video recordings drops to 120 GB. As will be presented in Section 3, the technique proposed in this paper allows to show to the user a set of previews. A single preview is an image representative of a whole video, so the compression is very high.

Real settings today are composed by many cameras. An interesting work that allows to monitor 20+ areas is DOTS [3], where the experience of a one-year installation of an indoor multi-camera surveillance system for use in an office setting is described. The user interface displays thumbnails of the cameras set in a building. The system tracks people movements and shows people positions in a 2D or a 3D model of the building. People are recognised through face-detection software. When the investigator is interested to one camera or one person, a preview area shows images coming from the selected camera. The system works by providing 2D and 3D models of camera settings, in this way the system can track people across multiple cameras. In the proposed version, MotionFinder addresses only one video at a time. DOTS could be inspiration for the next version of the tool.

A frequent requirement is to count people entering and exiting from a place. This problem has been faceted for various scenarios, like people going in and out a building, or a bus or a train [1]. Sidla et al. adopt a vision-based pedestrian detection and tracking system, able to count people in very crowded situations, like escalator entrances in under-

ground stations [11]. Our tool provides a simple counter that counts the different time intervals in which the movements have been occurred.

Current technologies allow people to track human gestures, and eye gaze, to help in human-computer interaction. Vural et al. uses eye-gaze analysis to capture overlooked actions in order to prepare a summarized video for a subsequent analysis [14]. This is particularly useful if there are many operators; the supervisor can quickly check what the operators observed.

Video summarization is an approach adopted in several systems. According to [9], each object moves in a "tube". By removing the constraint of the time in which an event occurred, it is possible to visualize several tubes in the same moment. The user perceives that many objects are doing something in the scene, this approach saves the human investigator time because, instead of watching moving things sequentially, the analyst can watch them in parallel. This approach is useful for videos of almost desert locations, but does not scales for crowded scenes.

A simple and effective technique is proposed by Tang et al. where the user draws small segments on the screen, the system shows the pixels beneath the segments in a timeline [12]. The idea behind this technique comes from how the scanner works. Only the parts of the video that "pass" through one or more segments drawn by the user are "scanned". For example, if such a line is drawn across a road, the resulting timeline will show the shapes of the car passing along it. These shapes will be more or less elongated depending on the speed. If nothing is happening, the line will keep writing the same pixels. So any object that passes over that line will be easily identifiable because background pixels, will be uniform and foreground objects will stand out. Our approach uses a similar idea but, instead of drawing a line, the user selects an area, and the tool shows videos, instead of still images.

3 The summarization technique

The approach presented in this paper relies on the creation of an image that summarizes the activity recorded by a stationary camera. The goal is to display traces of movements across the scene. The tool is very useful when it is known what happened, but the precise details about how and when are unknown (i.e. a laptop is stolen but it is not known when and by who). Traces of movements that took place in a given scene are shown using a color scale which ranges from yellow to red: yellow being used for previous activities, red used for more recent activities.

The video analysis process starts with the creation of an N x M matrix, called sequences matrix. N and M are the rows and columns of the original video frames respectively, so the matrix has the dimensions of the video resolution, expressed in pixels. Each cell of the matrix represent a corresponding pixel position in the video and contains a counter and a list of integers representing the frame number in which a movement was registered in the corresponding pixel; this list is called sequence. The counter is an integer indicating the number of frames in the sequence, see Figure 1; it represents the amount of activity for the associated pixel. The information about the amount of activity in a given pixel can be used t highlight "most active" pixels or filter out areas with a low activity or vice versa. Another use of the counter is that to quickly jump to the last recorded frame in order to get "the age" of the last movement in that pixel.

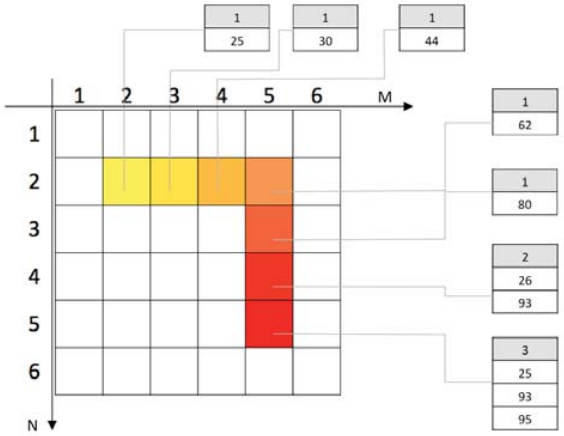

Figure 1. Simple example of generating the heat map from the sequence matrix

The proposed technique creates a summarization image whose goal is to show the activities occurred in the pixels of a video scene. The image is generated fom the matrix by assigning to each pixel a color using the yellow-red scale: a pixel showing a yellow color means that the last activity in that pixel occurred further back in time, while a pixel with a red color means that the last activity occurred recently. The resulting image shows all the movements that occurred in that certain scene together with a rough indication of how recent are the movements. If no movement is detected in the whole video in a specific pixel, the corresponding sequence will be empty and the pixel color in the visualized heat map will be transparent.

In order to better explain how the heat map is generated from the sequence matrix, Figure 1 shows a very simple example. In this exampe, the video has a resolution of 6 x 6 pixels, so the matrix will have 6 x 6 cells. Only in seven of the 36 cells something has moved, and the corresponding

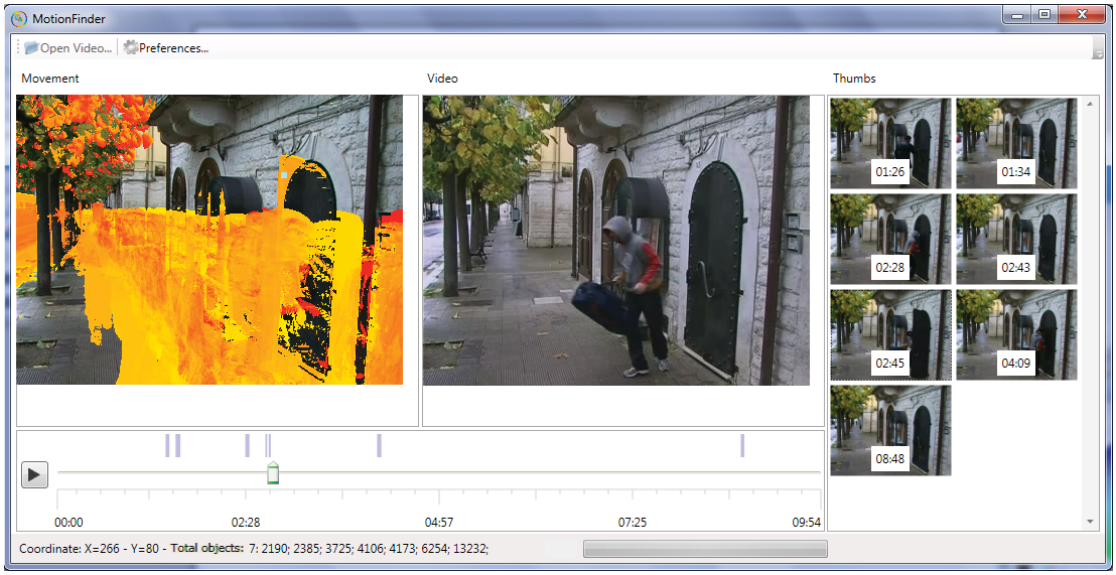

Figure 2. A screenshot taken from the analysis of the camera theft scenario

sequences are displayed. In this example, movements have been detected starting from the frame 25 until the frame 95. The cell (2,2) shows that the event occurred very early (in frame 25), while in cells (4,5) and (5,5) last activity occurred late in the video (in frames 93 and 95).

The video is analysed using the frame difference algorithm, which detects the movement of objects across two (single difference) or three frames (double difference [7]). Since the video sequences can be very long, a logarithmic scale can be applied in order to make visible movement that take place in short intervals of time.

4 MotionFinder

This section describes MotionFinder, a tool which performs video analysis by implementing the summarization technique we have illustrated in the previous section. MotionFinder is implemented in C#, and includes the OpenCV library for the image processing and object tracking. Since OpenCV is created in C/C++, the wrapping with C# is performed using the EmuCV platform.

The user interface of MotionFinder is shown in Figure 2. The image on the left is called *Movement* and shows the heat map produced from the sequence matrix as described in the previous section. The heat map represents, in a scale from yellow to red, traces of movements that took place in the scene; it is displayed overlayed on a frame in which no movement occurred. Yellow and red have been chosen for the heat map because, in outdoor environments, yellow and red are the least likely colors to be found; moreover, most camera equipments record video in grey scale. The heat map in Figure 2 that uses these two colors is well visible. Furthermore the yellow-red gradient carries some implicit meanings, like in thermography imaging, in which yellow is used for colder temperatures while red is used for hot temperatures. We consider an analogy to a heating process, in which hot temperature are reached during time; similarly, the map representing scene movements shows in yellow the previous movements and in red the most recent ones. The user may change the standard colors if desired or may desaturate the background color. In order to make visible the portion of the image occluded by the heat map, the user can make the map semi-transparent, as shown in Figure 3, or hide it temporarily.

The *Movement* area is interactive and the user can click in any part of it. By clicking on areas that do not contains any movement, nothing happens. If the user clicks on pixels with some movement, in the right area, called “Thumbs”, all the video excerpts corresponding to selected pixels appear. The Sequence matrix returns an ordered list of all frames related to the selection; MotionFinder splits this list into different frame sequences in order to build a number of excerpts representing different events.

In Figure 2, the information that seven parts of the video have been detected is provided in several ways: seven thumbnails appear in the “Thumb” area, at the right of the window; it is explicitly written in the status bar, at the bottom of the window; seven markers are displayed in the timeline, at the middle of the window.

Over each thumbnail is indicated the starting time of that specific video segment. It is possible to configure MotionFinder to read metadata about the video and show real date and time in which events occurred, but often this information is not explicitly provided; it is often coded in the video file name, and the creation date and time is not always reliable. In the case of Figure 2 metadata were not provided.

Below the "Movement" and "Video" areas, there is a timeline, which displays all detected excerpts as markers. If the user clicks on the "arrow" button positioned on the left of the timeline, all videos are played sequentially in the "Video" area. The user can otherwise start a single excerpt by clicking on the desired thumbnail in the "Thumbs" area; in that case only the clicked video is played.

The user may click on several points of the "Movement" area; points can be added or removed according to the user's needs. The sequence matrix, that holds information about where and when something happened allows the tool to respond in real-time. There are some hints that can be followed in the interaction with the heat map, in order to further speed up the analysis. One of the cases in which MotionFinder can be profitably used is when, before and after the event, the scene does not change, for example in the case of a door, or a window, that is opened and closed again; this action usually last for a few seconds, so other techniques, like jumping back and forth or fast-forwarding, are not effective. By clicking on the traces of the door that opens, like the click shown in Figure 2, the frames retrieved are those that belongs to the action, plus a number of frames before and after the movement. By default MotionFinder adds one second before and one second after, but this parameter can be changed.

5 An interaction example

In the context of video surveillance, someone may have stolen a valuable object or vandalized a store, etc. The human investigator who has to analyze the video has no clue about the specific time and how the event happened. One or more stationary cameras recorded the event. In order to speed up the search, motion detection sensors can be installed into the cameras, and may provide some help, this is very effective in mostly desert areas, but the case of a streets, corridors, with several side access or in frequently accessed areas, motion detection sensors are not really useful. The human investigator should review hours of video recordings in the hope of identifying the exact moment the event occurred.

We now describe how the tool shown in Figure 2 is used in order to support the human investigator to analyze the video and identify the frames related to that specific event. The video reported in the example of Figure 2 is recorded from a surveillance camera observing a street in which a photographer has his photography studio. The video have been recorded while the photographer was working in the studio.

Figure 3. The camera theft scene with the heat map overlay set to semi-transparent

During the day, he leaves the studio and close the door without locking it. When the photographer returns, he readily notices that an expensive video camera has been stolen. He does not know who, how and when exactly the theft took place but he assumes that something must have happened while he was away. He retrieves the video recording from the surveillance system and gives it as input to MotionFinder. The system produces the summarization shown in Figure 2.

Since the street is fairly trafficked, there is a lot of activity shown in the output. The photographer supposes that the thief must have entered the studio by the door. Thus, he clicks on traces of the upper corner of the door. The system identifies several moments in which the door was opened or closed. The timeline in the bottom part of Figure 2 shows seven markers placed at the corresponding times (also indicated in the status bar at the bottom of the window). In the right part of the screen a thumbnail appears for each detected excerpt. The photographer proceeds looking to the thumbnails and watching the video associated to those who seems suspicious.

He quickly identifies the moment in which he left the store, according to the thumbnails, at the time 2.28 the theft enters in the photography studio and exits at the time 2.45. The first thumbnail shows a person entering the studio and the next one after shows the previous person leaving while holding a bag with the stolen camera. Having watched the video excerpts and confirmed his suspicions, he exports the video excerpts in which the thief is shown entering and then

leaving so that he may deliver them to the authorities. Hopefully the police will be able to identify the thief.

6 Conclusions and future work

In this paper, MotionFinder, a tool for analysing video recorded by stationary surveillance cameras is presented. The tool supports the human investigator in the search for a particular event of which it is not known exactly when and how it occurred. MotionFinder summarizes the video into an interactive single image, composed by a background and a heat map showing traces produced by the movement of objects and people in the surveilled area. By interacting with this image the human investigator can see a set of excerpts related to the selection s/he made and, in seconds, get to the desired results.

One limitation of the adopted technique is about crowded scenes. The approach adopted by Pritch et al. [9] could be used to reduce the problem, but the heat map cluttering problem remains. One of the future directions could be to consider activities during time as different layers, like in the proposal of Romero et al. [10]. By providing the user with an interface for choosing the layers of interest, instead of querying the whole video the analysis may be speeded up.

During some informal evaluation sessions, some users suggested that it might be useful to incorporate the predominant direction of the movements.

7 Acknowledgments

Partial support for this research is provided by the VisMaster Coordination Action in the Future and Emerging Technologies (FET) programme under FET-Open grant number 225429.

References

[1] A. Albiol, I. Mora, and V. Naranjo. Real-time high density people counter using morphological tools. *Intelligent Transportation Systems, IEEE Transactions on*, 2(4):204 –218, Dec. 2001.

[2] P. Buono and A. L. Simeone. Video abstraction and detection of anomalies by tracking movements. In *International Conference on Advanced Visual Interfaces*, AVI '10, pages 249–252, New York, NY, USA, 2010. ACM.

[3] A. Girgensohn, D. Kimber, J. Vaughan, T. Yang, F. Shipman, T. Turner, E. Rieffel, L. Wilcox, F. Chen, and T. Dunnigan. Dots: support for effective video surveillance. In *15th international conference on Multimedia*, MULTIMEDIA '07, pages 423–432, New York, NY, USA, 2007. ACM.

[4] M. W. Green. The appropriate and effective use of security technologies in u.s. schools. a guide for schools and law enforcement agencies. Technical report, Sandia National Labs., Albuquerque, NM., 1999.

[5] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Haas, M. Lu, H. Merkl, and S. Pankanti. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. *Signal Processing Magazine, IEEE*, 22(2):38–51, March 2005.

[6] L. Huston, R. Sukthankar, J. Campbell, and P. Pillai. Forensic video reconstruction. In *ACM 2nd international workshop on Video surveillance & sensor networks*, VSSN '04, pages 20–28, New York, NY, USA, 2004. ACM.

[7] Y. Kameda and M. Minoh. A human motion estimation method using 3-successive video frames. In *VSMM'96, International Conference on Virtual Systems and Multimedia*, pages 135–140, 1996.

[8] D. A. Keim. Visual exploration of large data sets. *Communication of ACM*, 44:38–44, August 2001.

[9] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg. Webcam synopsis: Peeking around the world. In *11th IEEE International Conference on Computer Vision, 2007*, ICCV '07, pages 1–8, October 2007.

[10] M. Romero, J. Summet, J. Stasko, and G. Abowd. Viz-a-vis: Toward visualizing video through computer vision. *IEEE Transaction on Visualization and Computer Graphics*, 14(6):1261–1268, November/December 2008.

[11] O. Sidla, Y. Lypetskyy, N. Brandle, and S. Seer. Pedestrian detection and tracking for counting applications in crowded situations. In *Video and Signal Based Surveillance, 2006. AVSS '06. IEEE International Conference on*, pages 70 –70, 11 2006.

[12] A. Tang, S. Greenberg, and S. Fels. Exploring video streams using slit-tear visualizations. In *International Conference on Advanced Visual Interfaces*, AVI '08, pages 191–198, New York, NY, USA, 2008. ACM.

[13] M. Valera and S. Velastin. Intelligent distributed surveillance systems: a review. *Vision, Image and Signal Processing, IEEE Proceedings*, 152(2):192–204, April 2005.

[14] U. Vural and Y. Akgul. Eye-gaze based real-time surveillance video synopsis. *Pattern Recognition Letters*, 30(12):1151 – 1159, 2009. Image/video-based Pattern Analysis and HCI Applications.