A Real-Time Smart Assistant for Video Surveillance Through Handheld Devices

Hao Kuang[1], Benjamin Guthier[1], Mukesh Saini[1], Dwarikanath Mahapatra[2], Abdulmotaleb El Saddik[1]
[1]MCRLab, University of Ottawa, Canada
[1]{hkuan041, bguthier, msain2, elsaddik}@uottawa.ca
[2]Department of Computer Science, ETH Zurich, Switzerland
[2]dwarikanath.mahapatra@inf.ethz.ch

ABSTRACT

In a remote surveillance system, a high resolution surveillance camera streams its video to a user's handheld device. Such devices are unable to make use of the high resolution video due to their limited display size and bandwidth. In this paper, we propose a method to assist the mobile operator of the surveillance camera in focusing on sensitive regions of the video. Our system automatically identifies relevant regions. We introduce a pan and zoom strategy to ensure that the operator is able to see fine details in these areas while maintaining contextual knowledge. Regions of interest are identified using foreground detection as well as face and body detection. The efficacy of the proposed method is demonstrated through a user study. Our proposed method was reported to be more useful than two comparable approaches for getting an understanding of the activities in a surveillance scene while maintaining context.

Categories and Subject Descriptors

I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms

Algorithms

1. INTRODUCTION

Remote video surveillance systems use high resolution cameras to monitor security-critical areas like a parking lot or the entrance to a home. The captured video is streamed to a remote place where an operator can view it using a handheld device. In such scenarios, it may be undesirable to transmit the full resolution video to a bandwidth-limited handheld device. The screen of the device may also be too small to distinguish important details in the full view of the scene. Zooming into certain regions on the other hand comes with the risk of missing critical events in non-visible parts of the video and context from the surrounding may be lost. The user of a remote surveillance system thus needs to make a compromise between zooming in on detail and obtaining contextual information. This form of manually controlling the zooming and panning is tedious.

MM'14, November 3–7, 2014, Orlando, Florida, USA.

ACM 978-1-4503-3063-3/14/11 ...$15.00.
http://dx.doi.org/10.1145/2647868.2655070.

We propose a system to assist a user in remote surveillance with a handheld device. If no explicit control input is given by the user, the system automatically selects a region of interest (ROI) to zoom into and periodically displays the complete view to provide the necessary context. This selection is performed on the surveillance site to save bandwidth for transmission. Throughout this paper, we use the term *sensitivity* to denote the relevance of a certain part of the scene to the surveillance task. Sensitive regions are identified in real-time by employing a foreground detection method and by detecting faces and human bodies. In addition to the computed sensitivity, we allow user-defined static sensitivity. A user may for example find an entrance to a building relevant even if there are currently no people walking by. User-defined sensitivity also allows to avoid non-security critical areas with large amounts of motion, like trees moving in the wind. Our assistant selects the ROI that currently has the highest combined sensitivity and zooms into it. After a certain amount of time, it zooms out again to provide context. It then continues with the next sensitive region. The sensitivity of a region that has already been viewed is penalized to avoid selecting the same region twice. This penalty is lessened over time to allow the ROI to be selected again.

A field of research that is related to our work is *video retargeting*. Techniques in this field adapt high resolution videos to devices with a small display size like smartphones. The general goal is to fit as much of the "important" content as possible into the retargeted video. Importance may be measured from low level features such as saliency, contrast and gradients, or from higher level processing like face detection. Kopf et al. [4] give an overview of the vast number of existing approaches. Due to space limitations, we can only focus on the most prominent ones here.

The method presented in [5] crops and scales the frames of a video based on the importance of the pixels. This is done in a way that balances the loss of detail that occurs when downscaling with the loss of content when cropping. The currently shown area can be moved over the input video to create artificial pans and cuts whenever appropriate.

Techniques based on *Seam Carving* (see for example [3]) detect partially contiguous seams of less important pixels inside a frame and remove them. The size of the video is

thus reduced by removing unstructured areas between the important objects. Similarly, approaches based on *Warping* (see [9]) subdivide video frames into a rectangular mesh grid and transform each cell non-uniformly. Cells containing high importance remain mostly unchanged while unimportant cells may be warped. The advantage of Warping and Seam Carving over cropping is that content may be removed from the inside of a frame as opposed to only from the border. However, they also change the content of the frame, which makes them unsuitable for surveillance. The distance between objects may change which may have a severe impact on the context.

In general, video retargeting techniques have a different focus and are thus not applicable to a surveillance scenario. This is due to the following reasons. Low-level visual importance is generally not a good indicator for sensitivity in surveillance. A highly structured but static background may be interesting to a casual viewer but may have little relevance for security purposes. Also, video retargeting may discard certain areas of a video entirely as long as the result is aesthetically pleasing. In surveillance, discarding content is unacceptable. However, preservation of content can be achieved at the expense of aesthetics, e.g., by introducing artificial zooms. Furthermore, the common assumption of availability of the entire video beforehand is not valid in a real-time surveillance scenario.

An approach that is specific to retargeting of real-time surveillance videos was presented in [2]. After detecting sensitive areas from the difference of two consecutive frames, the video is cropped and zoomed to the target size using a moving ROI. Artificial cuts may be introduced if necessary. The work is based on a scenario where the retargeted video is shown in a small area on a big screen, where an overview of the entire scene is also available. This is not the case when showing the video on a mobile device. Furthermore, the simple form of sensitivity map that is being used without considering user preferences may lead to a bias towards moving high-contrast objects, even when they are not relevant to security (e.g., a flag waving in the background).

2. VIDEO SURVEILLANCE ASSISTANT

Our assistant for video surveillance on handheld devices consists of three components that work together: Sensitivity map computation, ROI selection, and presentation. It uses a buffer of ω frames which allows it to look into the future when selecting an ROI. This introduces a latency of ω frames, which is in the order of five to ten seconds. In a remote surveillance system, where the user is typically far away from the site under surveillance, such a latency is uncritical. Note that the buffer is located at the surveillance site; no buffer is required on the mobile device.

The sensitivity map component obtains the latest frame from the camera and calculates a sensitivity map $\mathcal{S}$ for the frame. Over the course of ω frames, the $\mathcal{S}$ are summed up to an accumulated sensitivity map $\mathcal{S}^\omega$ which is then passed on to the ROI selection component. We use the ω superscript to denote maps that are used only once every ω frames. The ROI selection component combines $\mathcal{S}^\omega$ with the static user input $\mathcal{U}$ and the penalty map $\mathcal{P}^\omega$. The result is a decision map $\mathcal{D}^\omega$ from which the ROI to be displayed is calculated. This ROI is passed on to the presentation component, which now processes the end of the video buffer. Over the duration

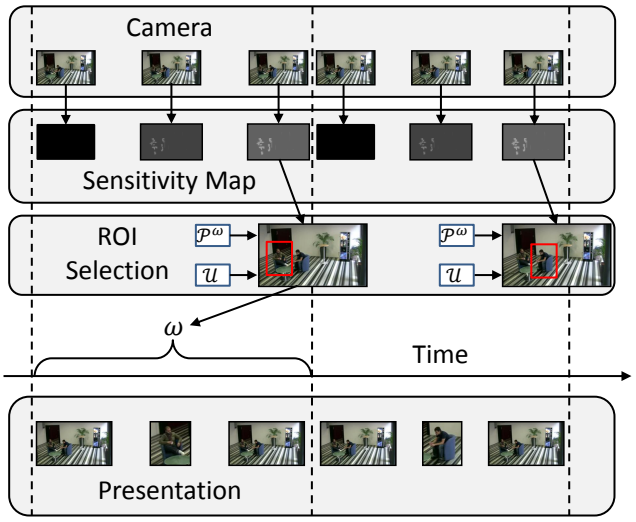

Figure 1: Overview of our proposed remote video surveillance assistant. A sensitivity map is calculated for every frame. It is accumulated over ω frames and then passed to the ROI selection. The sensitivity, penalty and user input maps are combined to select a ROI. The presentation component operates with a delay of ω frames. It zooms into the ROI over the course of ω frames.

of ω frames, the presenter smoothly zooms into the selected ROI and zooms out again. The resulting resized frames are sent over the network to the handheld device. See Figure 1 for an overview of the system.

The sensitivity map component and the presenter operate in sync with each other and the frame rate of the camera, but with ω frames delay between camera and presentation. ROI selection is triggered each time ω frames have been accumulated. Details on the three components is given in the following sections.

2.1 Sensitivity Map

In a surveillance scenario, humans are generally the main focus of attention. In order to recognize humans in a video, further attention must be paid to their faces. In addition to that, objects moving relatively to the static background, i.e. foreground objects, are also interesting. Our sensitivity map is thus computed from the results of face detection, human body detection and foreground object detection.

We begin by obtaining the latest frame from the camera and storing it in the buffer for later use by the presentation component. The approaches [8] and [1] are then used on the frame to detect faces and human bodies, respectively. Faces and bodies are represented by bounding boxes containing them. From these, we create two binary images I_f and I_b that are 1 in areas where faces and bodies were detected and 0 otherwise. Moving foreground objects are detected using the Gaussian mixture model presented in [7]. The result is a binary image I_o with pixels belonging to a moving foreground object being 1. The sensitivity map $\mathcal{S}$ for the frame is then calculated as $\mathcal{S} = I_f + I_b + I_o$. Calculating $\mathcal{S}$ like this automatically prioritizes faces over human bodies

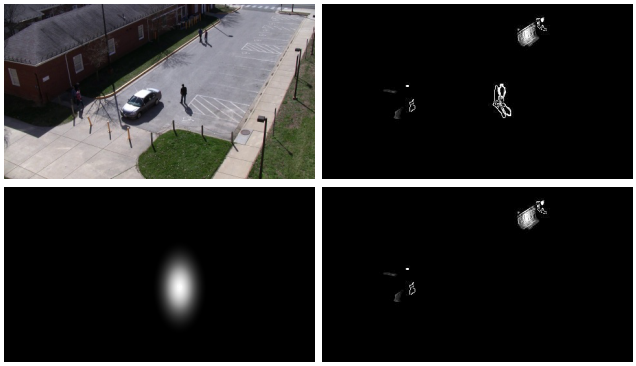

Figure 2: The top left image shows a frame from a surveillance video. The top right frame shows the accumulated sensitivity map $\mathcal{S}^\omega$. The person in the center gets chosen as the next ROI. As a result of the chosen ROI, the penalty map on the bottom left gets updated. After ω frames, the previously chosen region gets penalized in the sensitivity map and will not be chosen again (bottom right).

(because $I_f \subseteq I_b$) and moving bodies over moving objects (when $I_b \subseteq I_o$). The face of a person walking through the scene for example would be included in $\mathcal{S}$ three times.

Our system needs to detect the area of the video with the highest sensitivity over the course of ω frames. For this reason, the sensitivity maps for each frame in this timespan are summed up into an accumulated sensitivity map $\mathcal{S}^\omega$. Accumulating the sensitivity also eliminates spuriously detected faces and human bodies and reduces the impact of noise in I_o. Once $\mathcal{S}^\omega$ has been accumulated over ω frames, it is passed on to the ROI selection component.

2.2 ROI Selection

The sensitivity map $\mathcal{S}^\omega$ as defined above cannot always identify the most important aspects of the scene. Importance also depends on context, which is difficult to assess algorithmically. In a hallway for example, the entrance area is more important to focus on than the walls. A surveillance video of a parking lot might also include parts of the adjacent road which could be outside of the scope of surveillance. It is thus important to include user input into the decision making process. We model the user input as a map $\mathcal{U}$ of the same size as $\mathcal{S}^\omega$ which gives an offset to the sensitivity. Since this information is mostly static, it needs to be defined only once when the system is set up.

In order to not display the same region over and over, we give a penalty to the sensitivity of a region that has been presented to the operator before. Our penalty map $\mathcal{P}^\omega$ has values between zero and one with higher values meaning a higher penalty. After having selected an ROI to display, we calculate a 2D Gaussian function for the chosen ROI. Its mean is the center of the ROI, and the variances are chosen relative to the width and height of the ROI. This Gaussian function is added to $\mathcal{P}^\omega$. The process of penalizing a previously chosen ROI is illustrated in Figure 2. Every ω frames when a new ROI is selected, the penalty map is multiplied by a factor $\alpha \in [0, 1]$ to decrease the penalty over time. The sensitivity of a selected ROI is thus reduced at first, and its sensitivity then slowly increases back to its original value.

When a new $\mathcal{S}^\omega$ is available – once every ω frames – it is combined with the user input and the penalty map to form a final decision map. The decision map $\mathcal{D}^\omega$ is defined as

$$\mathcal{D}^\omega = (1 - \mathcal{P}^\omega) \cdot (\mathcal{S}^\omega + \mathcal{U}). \quad (1)$$

All operators used here process the maps pixel-wise.

For the purpose of ROI detection, $\mathcal{D}^\omega$ is converted into a binary image by applying a low threshold. Morphological operations are then used to reduce noise. Next, the system extracts contours from white areas in the binary image and merges adjacent contours if necessary. The bounding boxes of the connected contours are the potential regions of interest to zoom into. For each potential ROI, we compute a decision value. It is calculated by summing up all values inside the ROI in the original unthresholded $\mathcal{D}^\omega$. This allows ranking of the potential ROIs according to the amount of sensitivity they contain. The ROI with the highest decision value is selected and sent to the presentation component.

2.3 Presentation

The presentation component creates the output frames that are sent to the remote operator. At the beginning of each block of ω frames, it is given an ROI for the entire block. The frames of the block are contained in the buffer. Note that the ROI selection may give rectangular areas with an arbitrary aspect ratio. The aspect ratio $\phi = w/h$ of the ROI is generally not identical to the target aspect ratio ϕ_t. $\phi < \phi_t$ means that the chosen ROI is too high. We thus increase the width of the ROI as $w = \phi_t h$, so that the aspect ratios match again. The case $\phi > \phi_t$ is handled analogously. This guarantees that the selected region is fully contained in the created view.

The presentation of a block of frames always starts and ends with a fully zoomed out overview. Within the block, it zooms into the ROI, stays zoomed in for a certain amount of time and then zooms out again. Zooming is implemented as an interpolation between the parameters of the full frame and the chosen ROI. We use cubic spine interpolation for smooth zooming. Like this, an interpolated ROI for the current frame is calculated. The presenter takes the oldest frame from the buffer and crops it according to the interpolated ROI. The cropped frame is then scaled to the target resolution and sent over the network.

3. EXPERIMENTAL RESULTS

The main goal of our approach is to resize the video from a surveillance camera so that it can be used for remote monitoring on a mobile device with a small screen size. We evaluated our approach in a user study with 22 male and 7 female non-expert subjects. The average age was 25 with a standard deviation of 4. Four surveillance videos with a resolution of 1920×1080 and a length of one minute were used. They were taken from the VIRAT database, which was designed for performance assessment of activity detection algorithms [6]. Representative frames from the four scenarios are shown in Figure 3.

As the target resolution, we chose 384×216, which is $1/5$ of the original resolution. Three different videos of the target resolution were created for each scenario: A scaled

Statement	Scaled	Pan Only	Proposed
1	3.1 (1.4)	3.3 (1.0)	4.5 (0.9)
2	2.6 (1.2)	3.3 (0.8)	4.3 (1.0)
3	2.6 (1.2)	3.2 (1.0)	4.5 (0.8)
4	3.7 (1.0)	3.1 (1.0)	2.1 (1.2)
5	2.7 (1.3)	3.3 (1.0)	4.4 (0.8)

Table 1: Agreement scores for the five statements and the three approaches. The values are averaged over the four scenes and all participants, with the standard deviation given in parentheses.

down version of the original video which served as a baseline for the evaluation ("scaled"), the video created by the proposed method ("proposed"), and a video that uses the proposed ROI selection, but only pans between the ROIs without zooming out to the full overview in between ("pan only").

The users were given a brief introduction. Their task was to assume the role of a security operator who monitors the area and detects abnormal activities. All three versions of the scenario were shown next to each other simultaneously. After watching the videos, the users were given five statements about each version. They had to rate each statement on a scale from 1 (strongly disagree) to 5 (strongly agree). The statements were:

1. the video provides full coverage of the site,
2. the video provides all details of the site,
3. the camera motion was helpful in monitoring the area,
4. the video is boring, and
5. it is easy to understand the activities in the video.

The website we used for the study can be found at [1].

Table 1 shows the results of our study. Since the users' ratings were similar across all four scenarios, we only show the averaged values here. From the table, it can be seen that our approach achieves better results than the two compared methods in all five considered categories. Note that for statement 4 (the video is boring), a lower value is better. Out of the two other approaches (scaled and pan only), the video that pans between the ROIs seems to be more useful. This is due to the small output resolution. The scaled version was too small to be useful for monitoring. In statements 1 to 3 and 5, our approach obtains an average score between 4.3 and 4.5. This indicates that the participants of the study generally found our retargeting approach to provide videos that were helpful in the given surveillance task.

4. CONCLUSIONS

We proposed a smart assistant for remote video surveillance on a handheld device. A high-resolution surveillance video was retargeted to a smaller resolution to be displayable on a small screen and save bandwidth. This was done by first identifying regions of interest in the video and then zooming into these regions one after another. An overview of the entire sceen is given periodically to provide the necessary context to the operator. The results of our user study show, that the proposed method provides full coverage of the scene while also showing the detail that is necessary for understanding the activities therein. The users generally reported that the smart assistant system was helpful in remote monitoring a scene under surveillance.

[1] https://sites.google.com/site/acm2014ssa/

Figure 3: Example frames from the four scenes used in the study. All are outdoor surveillance scenes and the camera is static. The fourth scene (bottom right) is a parking lot. Here, the adjacent road was out of the scope of surveillance. By specifying user input, the motion outside the parking lot is ignored when selecting an ROI.

5. REFERENCES

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of the Computer Vision and Pattern Recognition*, volume 1, pages 886–893. IEEE, 2005.

[2] H. El-Alfy, D. Jacobs, and L. Davis. Multi-scale video cropping. In *Proc. of the 15th int. conf. on Multimedia*, pages 97–106. ACM, 2007.

[3] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Discontinuous seam-carving for video retargeting. In *Proc. of the CVPR*, pages 569–576. IEEE, 2010.

[4] S. Kopf, T. Haenselmann, J. Kiess, B. Guthier, and W. Effelsberg. Algorithms for video retargeting. *MTAP*, 51(2):819–861, 2011.

[5] F. Liu and M. Gleicher. Video retargeting: automating pan and scan. In *Proc. of the 14th int. conf. on Multimedia*, pages 241–250. ACM, 2006.

[6] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *Proc. of the CVPR*, pages 3153–3160. IEEE, 2011.

[7] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. of the CVPR*, volume 2. IEEE, 1999.

[8] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of the Computer Vision and Pattern Recognition*, volume 1, pages I–511 – I–518. IEEE, 2001.

[9] Y.-S. Wang, J.-H. Hsiao, O. Sorkine, and T.-Y. Lee. Scalable and coherent video resizing with per-frame optimization. In *ToG*, volume 30, page 88. ACM, 2011.