Object Video Streams: A Framework for Preserving Privacy in Video Surveillance

Faisal Z. Qureshi

Abstract Here we introduce a framework for preserving privacy in video surveillance. Raw video footage is decomposed into a background and one or more object-video streams. Such object-centric decomposition of the incoming video footage opens up new possibilities to provide visual surveillance of an area without compromising the privacy of the individuals present in that area. Object-video streams allow us to render the scene in a variety of ways: 1) individuals in the scene can be represented as blobs, obscuring their identities; 2) foreground objects can be color coded to convey subtle scene information to the operator, again without revealing the identities of the individuals present in the scene; 3) the scene can be partially rendered, i.e., revealing the identities of *some* individuals, while preserving the anonymity of others, etc. We evaluate our approach in a virtual train station environment populated by autonomous, lifelike virtual pedestrians. We also demonstrate our approach on real video footage. Lastly, we show that Microsoft Kinect sensor can be used to decompose the incoming video footage into object-video streams.

1 Introduction

Video surveillance is ubiquitous. Recent advances in camera and communication technologies along with the decrease in deployment costs have made it possible to set up large video surveillance infrastructures relatively easily. The societal shift that has occurred during the first decade of the 21st century with its focus on the *war on terrorism* has all but removed any opposition to putting citizenry under video surveillance with the stated aim to enhance public safety and security. Many cities around the world are increasingly relying on video surveillance for crime prevention and community safety. Video footage captured through surveillance cameras

Faisal Z. Qureshi
Faculty of Science, University of Ontario Institute of Technology, Oshawa ON Canada, e-mail: faisal.qureshi@uoit.ca

is routinely used to identify suspects and as evidence in the courts. In addition to the video surveillance infrastructure controlled by city councils and government bodies, private sector has also invested heavily in video surveillance technologies. Retail stores, for example, are using video cameras to collect data needed to analyze and model consumer behavior [14, 11]. Video cameras are also quickly becoming an essential part of smart environments, e.g., supporting home automation to enable elderly and disabled to safely remain in their own homes.

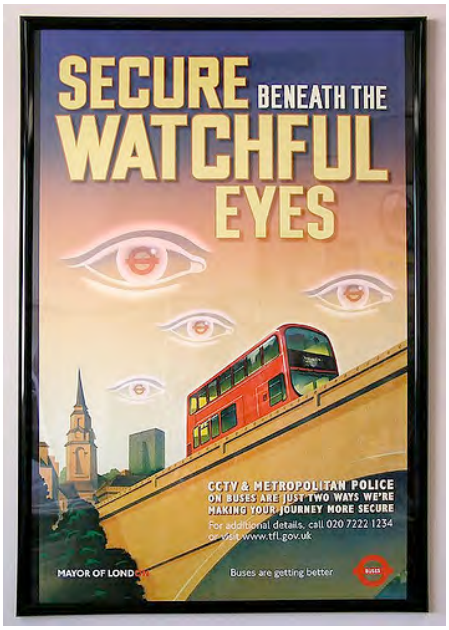


Fig. 1: British Government poster outside Metro station in London (*circa* 2007).

The panoptic effect of pervasive video surveillance (Fig. 1) raises many questions: (1) Who is collecting information about us? (2) How this information is being used? (3) What information is being collected? (4) Who has access to this information? and (5) What is the retention policy for the collected information? These issues have been studied by social and legal experts, and policies and best practices have been suggested. The use of video surveillance, however, is still largely unregulated. In 2001 Superbowl, law enforcement videotaped attendees without their knowledge, and then compared their faces against a database containing faces of known criminals [13]. Casinos, for example, also use biometric technology to identify cheaters and for "patron management" [12]. Experts agree that video surveillance undermines our "right to anonymity." Video surveillance augmented with biometric technology (e.g., face recognition) raises even more privacy concerns. Balancing the need for video surveillance against an individual's right to privacy is a challenge that needs to be addressed within social, legal, and technical contexts. A timely

challenge for computer vision researchers is to develop video surveillance systems with built-in *privacy protection* capabilities. Such capabilities will help camera operators implement best practices and uphold laws regulating video surveillance.

Here we introduce a framework for privacy preserving video surveillance systems.[1] Captured video is decomposed into *object-video* streams. Each object-video stream contains visual information about a single object in the scene.[2] These streams can be recombined to visualize the area under surveillance in a variety of ways. For example, individuals present in the scene can be represented as color-coded blobs, hiding their identities. Selected individuals can be also blurred. Additionally some individuals can be removed from the video entirely. We also envision that these object-video streams are encrypted at source and can only be viewed by operators with the necessary authorization.

We embrace the *Virtual Vision* paradigm, exploiting visually and behaviorally realistic virtual environments to develop and empirically evaluate our video surveillance framework [16]. We employ a virtual train station environment populated by autonomous lifelike virtual pedestrians that is described in [24]. The vision pipeline for our prototype video surveillance system matches the performance of the vision pipeline (for real video) presented in [7]. Therefore, the obtained results are legitimate and valuable. We describe vision pipeline in Sec. 3. We also show object-stream construction and selective rendering using real video footage in Fig. 9. Furthermore, we show decomposing video into object-video streams using the Microsoft Kinect sensor [1].

The remainder of the chapter is organized as follows. We summarize relevant literature in the next section. Sec. 3 develops the vision pipeline: background learning, foreground detection, and pedestrian tracking. Then in Sec. 4, we describe how raw video is decomposed into a background stream and one or more object-video streams. Sec. 5 describes how object-video streams can be used to develop a privacy preserving video surveillance system. Preliminary results of our approach are presented in Sec. 6. While we have not deployed and tested the our system in a real-world setting, the results presented here serve to demonstrate the applicability of the proposed strategy. We conclude our chapter with conclusions and future directions in Sec. 7

2 Relevant Literature

Typically, sensory data gathered by a video surveillance system is monitored by human operators to detect events of interest. Computer vision technologies, such

[1] This chapter is based upon our paper that appeared in the 6th International Conference on Advanced Video and Signal Based Surveillance in 2009 [17].

[2] This assumption sometimes breaks due to the limitations of video processing routines, such as background subtraction, object tracking, image segmentation, etc. Still under favourable conditions—good lighting, sparsely populated scenes, etc.—it is possible to decompose the video into object-video streams as we show later in the chapter.

as pedestrian tracking, face recognition, and detection of unclaimed baggage, have been employed to increase the effectiveness of existing video surveillance systems and to develop the next-generation camera networks capable of perceptive coverage of large areas with little or no human supervision. These highly capable video surveillance systems shift the balance of power between intrusiveness and privacy, raising new privacy concerns. Clearly, these systems severely undermine the right to anonymity in public space.

The ability to visually track people present in the scene is necessary for camera networks capable of carrying out visual surveillance tasks autonomously. Face detection and recognition enable these networks to identify individuals [26, 8, 4, 28, 2]. Computer vision techniques also allow these video surveillance systems to compute soft and hard biometric signatures of individuals. In short, computer vision technologies will play a central role in developing the video surveillance systems of the future.

Interestingly computer vision technologies can also be used to develop camera networks that can uphold privacy policies and regulations [22, 6]. Pedestrian detection and tracking routines can identify individuals present in the scene and obscure them to hide their identities. The operator can still see the scene and know how many people are present in the scene without knowing the identities of those people. An activity recognition technique can reveal an individual if it detects an anomalous behavior.

Schiff *et al*. develop a video surveillance system capable of obscuring the faces of individuals present in the scene [21]. Individuals who do not want to be identified wear a visual marker, which allows the video surveillance system to locate the face of the individual and obscure it with an ellipse, while allowing observation of his or her actions in full detail. This allows the operator to observe the activities taking place in the scene without knowing the identities of the people present.

Sony patented a privacy mode for camcorders that replaces the skin color of individuals so as to avoid race-based discrimination [3]. [27] patented a system capable of obscuring a privacy region in a pan-tilt-zoom camera. [9] develops a system that is able to locate and obscure people in a video, thereby preventing statistical inferences from the video. Chattopadhyay and Boult developed a privacy preserving smart camera, called *PrivacyCam* [6]. PrivacyCam uses on-board digital signal processor to locate and encrypt human faces in the image. The original image can be recovered given the correct decryption key.

Saini *et al*. have carefully studied privacy leakage in video surveillance systems [18]. They correctly identify that an individual's identity can be learned through other channels even when that individual is not identifiable within a video. Consequently, obscuring/blurring an individual in a video footage alone is not sufficient to ensure that the privacy of that individual is not compromised. Object-video streams might alleviate this problem somewhat, since it is possible to make an individual disappear from the video by simply removing the object-video stream corresponding to that individual from the mix. Saini *et al*. have studied adaptive video blurring to protect the privacy of individuals present in the scene [19].

3 Vision Pipeline

The performance of the proposed surveillance system is ultimately tied to the capabilities of the vision pipeline that is responsible for segmenting raw video into object video streams. We have adapted well-understood computer vision algorithms, including background subtraction, blob detection, and pedestrian tracking, to construct a vision pipeline that works equally well on both synthetic video captured within our virtual vision simulator and real video captured by physical cameras. Recently, we have also used the Microsoft Kinect RGBD sensor to construct object video streams from raw videos. Below we briefly explain the various components of the vision pipeline.

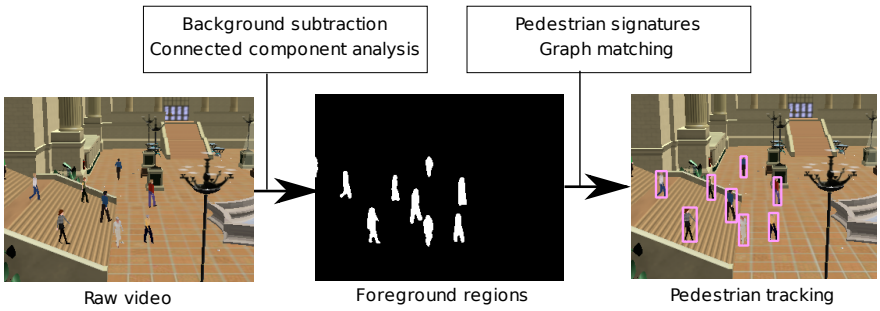


Fig. 2: Vision pipeline: We have adapted well-understood computer vision algorithms for our purposes. The vision routines operate upon both synthetic video captured by virtual cameras and real video captured through physical cameras. Background subtraction is used to identify foreground pixels. Pedestrians signatures that encode pedestrian color distribution in HSV space are matched in successive frames to perform tracking.

3.1 Background Subtraction

During an initial training phase, when no pedestrian is visible, each camera learns a background model of the scene. We model the variation in each pixel using the codebook method that was developed in [10]. We use the implementation of codebook method for background learning provided in the Open Computer Vision Library (OpenCV) [5]. Background subtraction step involves comparing the current frame against the learnt background model and constructing a (in general, noisy) foreground mask. In our case, the foreground mask constructed through background subtraction is cleaner due to lack of shadows, however, this does not invalidate our vision pipeline. Many techniques exist in the literature to account for shadows and

other artifacts, such as camera motion, during background subtraction [7]. In a real system, we would also need a mechanism to update the background model to account for changes in the background. It is straightforward to incorporate this capability into our background model.

3.2 Pedestrian Tracking

The foreground mask obtained through background subtraction is cleaned up through connected component analysis and blobs representing foreground objects are extracted. In our case, each blob represents one or more pedestrians. We employ an appearance-based pedestrian tracker that is able to detect and track pedestrians in both synthetic and real video footage. Pedestrian appearance signatures are matched across frames to track pedestrians. Specifically pedestrian tracking is performed by setting up a bipartite graph matching problem as suggested in [7]. The optimal solution to the matching problem resolves pedestrian identities across multiple frames. We refer the reader to [7] for more details. Pedestrian tracker assigns each blob to one or more pedestrians. If an appropriate blob is not found in a frame, the pedestrian is matched to the entire frame.

The tracker maintains a list of pedestrians that are currently being tracked. In each frame, each pedestrian is either matched to a blob (using pedestrian signature matching) or to the background. The tracker is robust to short-duration occlusions.

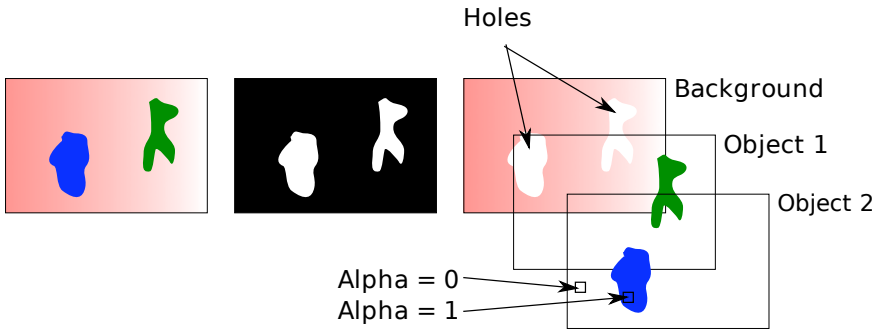


Fig. 3: Cleaned up foreground mask decomposes a video frame into a background component and two foreground components. Pedestrian to blob mapping information maintained by the pedestrian tracker links each foreground component to one (or more) pedestrians.

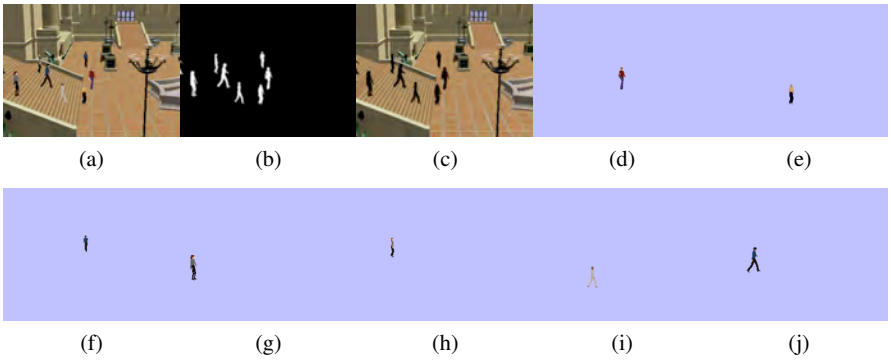


Fig. 4: Decomposing video into a background component and 7 foreground components. Each foreground component encodes visual data for a particular pedestrian. (a) Raw video. (b) Foreground mask. (c) Background image containing holes. (d)-(j) RGBA frames containing color data for 7 pedestrians visible in the frame.

3.3 Microsoft Kinect RGBD Sensor

It turns out that Microsoft Kinect Red-Green-Blue-Depth (RGBD) sensor is able to perform background subtraction, blob detection, pedestrian tracking, and pose estimation in real-time (around 15 frames per second). Furthermore Microsoft Kinect also estimates the 2.5D structure of the scene by associating a depth value with each pixel. The depth information makes it much easier to identify the blobs belonging to different individuals present in the scene, which is the first step towards constructing object video streams from raw videos. In other words Microsoft Kinect already includes the vision pipeline that we require. It is, however, important to bear in mind that the Kinect sensor's operational range is limited to roughly 2.5m. Consequently we still need our vision pipeline in order to be able to use generic cameras that have much larger operational ranges.

4 Object-Video Streams

This section describes the process of decomposing captured video into object-video streams. Let F_t be the video frame and M_t be the (binary) foreground mask at time t. We begin by extracting background pixels:

$$F_t^B(\mathbf{x}) = \begin{cases} [F_t(\mathbf{x}), 1] & \text{if } M_t(\mathbf{x}) = 0; \\ \mathbf{0} & \text{otherwise.} \end{cases} \quad (1)$$

Here, $\mathbf{x}$ is defined over the domain of F_t. $[F_t(\mathbf{x}),1]$ denotes an RGBA vector and $\mathbf{0}$ denotes a zero vector. F_t^B is an RGBA image. Next, assume that the foreground mask M_t contains n blobs. Then for each blob C_i identified in the foreground image F_t, perform the following steps,

1. Construct blob mask M_t^i.

$$M_t^i(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in A(C_i); \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

   $A(C_i)$ denotes the area enclosed by blob C_i.
2. Construct an RGBA color image F_t^i.

$$F_t^i(\mathbf{x}) = \begin{cases} [F_t(\mathbf{x}),1] & \text{if } M_t^i(\mathbf{x}) = 1; \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{3}$$

   Here, $\mathbf{x}$ is defined over the domain of F_t. $[F_t(\mathbf{x}),1]$ is an RGBA vector. $\mathbf{0}$ denotes a zero vector.

The above process, which is illustrated in Fig. 3 and 4, partitions frame F_t into a background image, F_t^B, (with holes in places of foreground objects) and n object images F_t^i, where $i \in [1,n]$. Each object image contains pixel data for one (or more) foreground objects. We note that this is a loss-less operation by observing that

$$F_t = F_t^B \cup (\cup_i F_t^i).$$

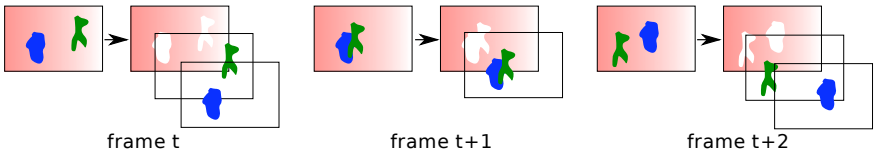
We define a $Partition(.)$ operator that partitions a frame into background and foreground components as described above:

$$Partition(F_t) = \{F_t^B, F_t^i | i \in [1,n]\}.$$

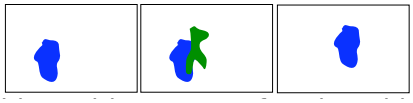
Given a sequence of video frames F_t, we construct the object-video stream O^k for a particular object k as follows. Let O^k be an empty sequence. Then for each frame F_t,

1. Construct $Partition(F_t)$.
2. Extend the sequence O^k by appending F_t^i at the end, if the tracker maps object k to blob i at time t. If the tracker does not map object k to any blob in the current frame, extend the sequence O^k by appending F^t.

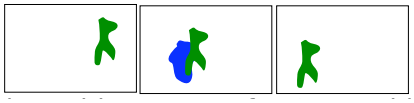
Pedestrian crossover, proximity or occlusions can lead to poor blob segmentation and tracking errors. Multiple pedestrians can be mapped to the same blob. Consider, for example, the scenario shown in Fig. 5. The two objects represented as Green and Blue blobs are correctly segmented in frame t, so frame t is correctly decomposed into three components: background, Blue object, Green object. In frame t+1, however, the two objects are seen as a single blob, and the frame is incorrectly decomposed into two components. The pedestrian tracker assigns both objects to Blue/Green blob. Next, the two objects are correctly segmented in frame t+2, so frame t+2 is correctly decomposed into three components.



(a) Raw video frame is decomposed into a background stream and foreground stream(s)



(b) Object-video stream for Blue object



(c) Object-video stream for Green object

Fig. 5: Constructing object-video streams.

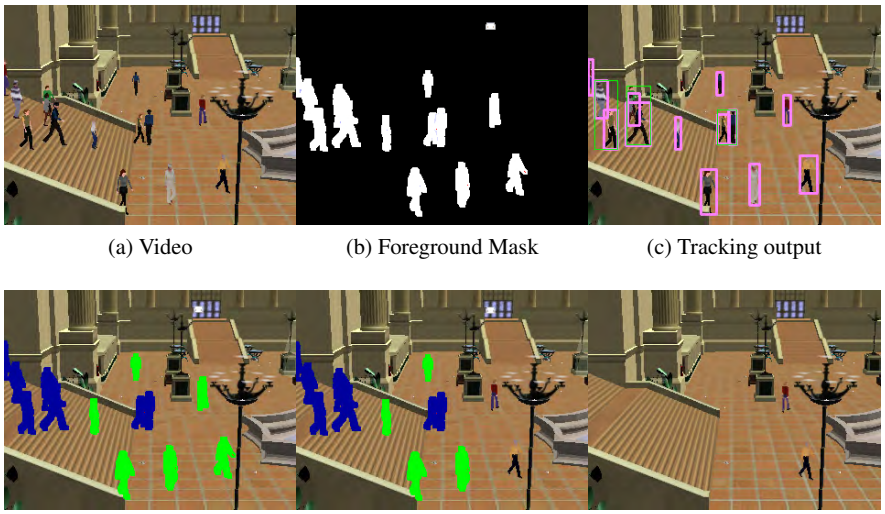
5 Privacy

Decomposing raw video into object streams opens up new possibilities for implementing privacy policies. At the most basic level, it allows the video surveillance system to obscure the identities of individuals present in the scene. An operator can still see scene activity without knowing the identities of individuals present in the scene. Object-video streams can be used to render the scene for a variety of purposes. We employ Laplacian pyramid blending to combine different object-video streams for rendering purposes [15]. Laplacian pyramid blending is also used to fill the holes in the rendered scene by using the stored average background image F^B.

- Object-video streams can be used to enhance the situational awareness of the operator. Objects can be color coded to convey qualitative scene information to the operator. This can be a powerful scheme for drawing operator's attention to events of interest. Sophisticated video analytics or simple image-space heuristics can assign unique colors to pedestrian blobs. For example, any pedestrian who enters a prohibited zone can be drawn as a red blob. Similarly, poorly segmented blobs, which map to multiple pedestrians, can be color coded to indicate pedestrian interactions (or simply overlap).
- Object-video streams also enable selective scene rendering. An operator can render the scene showing only some of the pedestrians present in the scene, without disclosing the identities of other individuals.
- Object centric decomposition of surveillance video has the potential to give more control to the individual. E.g., a person might be able to find a lost item by sifting through an appropriate rendering of the scene that hides the identity of other individuals. Presently individuals are not allowed the access to the surveillance video as it might violate the privacy of others present in the scene.

We will be remiss to not point out that similar ideas of leveraging computer vision to obscure the identitiy of individuals present in the scene have been explored

by others [23]. It is envisioned that in a real video surveillance system, object-video streams will be encrypted. Access control mechanisms will determine how the scene is rendered providing a way to strike a balance between the need-to-know on the part of an operator and the right-to-privacy on the part of an individual.



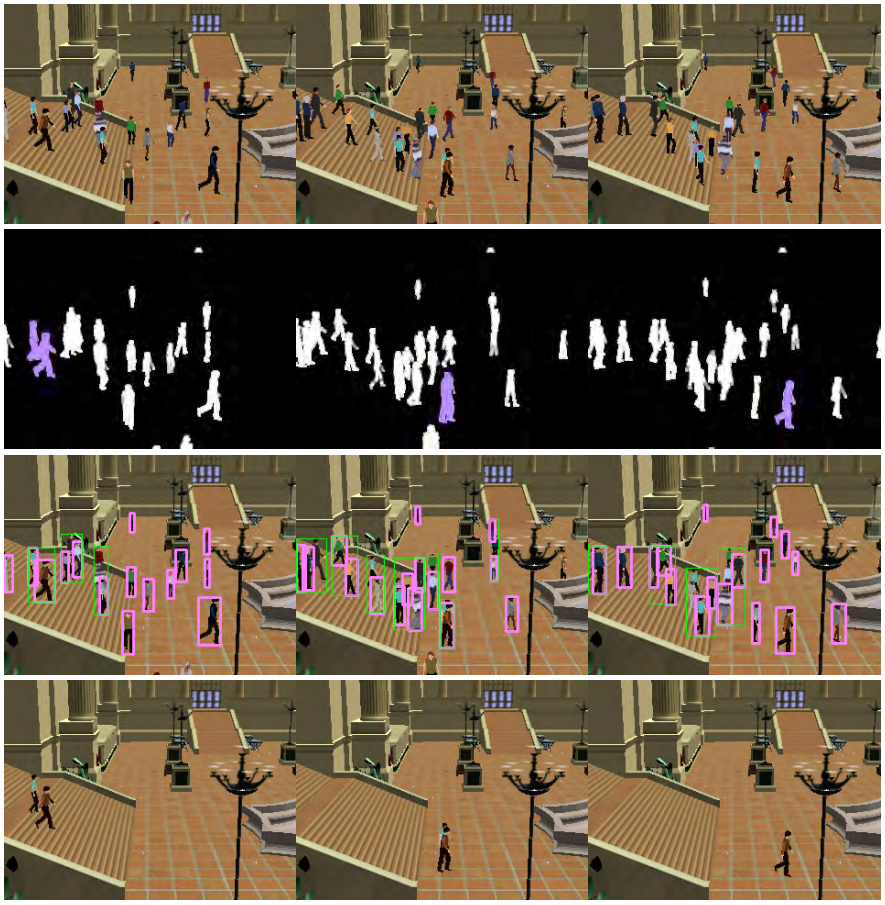
(a) Video (b) Foreground Mask (c) Tracking output

(d) Rendering 1 (e) Rendering 2 (f) Rendering 3

Fig. 6: Decomposition into object-video streams presents new possibilities to view the scene.

6 Results

We evaluate our approach on a *virtual* video surveillance system deployed in a virtual train station. The video surveillance system comprises 4 passive, wide field-of-view cameras with overlapping fields-of-view. It is assumed that the camera setup is fully calibrated, which simplifies pedestrian identity management across multiple cameras. Decomposing raw video into object-video streams does not require the camera network to be calibrated. We also report results on real video footage, further demonstrating the validity of our approach. Lastly we demonstrate how Microsoft Kinect RGBD sensor can be used to construct object-video streams.

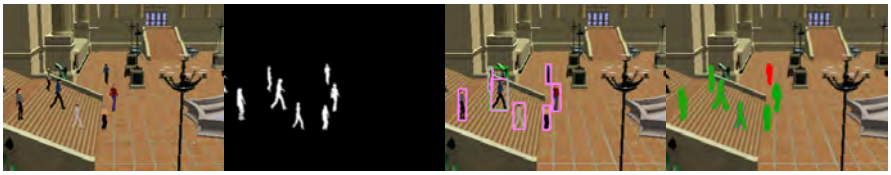


Frame 266 Frame 308 Frame 325

Fig. 7: This sequence shows the effects of poor foreground segmentation on the object-video stream for the pedestrian wearing a Brown shirt. Pedestrian tracker maps the pedestrian of interest to Violet blobs in the shown frames.

6.1 Synthetic Footage

We show different rendering possibilities in Fig. 6. Fig. 6(d) shows a privacy preserving rendering where each pedestrian is seen as a color blob. Single person blobs are Green; whereas, multi-person blobs are colored Blue. Pedestrian tracker selects an appropriate color for the blob. Fig. 6(e) shows a rendering where the identities of two individuals (the man in Red shirt and the man in Orange shirt) have been revealed. All other individuals are still shown as blobs. Fig. 6(f) is showing the



(a) (b) (c) (d)

Fig. 8: Event based color coding is also possible. The Red blob indicates a person who has tripped a virtual wire (defined in pixel space). Such wires are routinely used in video surveillance systems. (a) Video frame, (b) foreground mask, (c) tracking output, and (d) privacy preserving color coded rendering.

scene with only two persons. In this case, the viewer can know the identity of these persons; however, he can not tell how many people were present in the scene.

Fig. 7 shows selective rendering. The top row contains original video frames. The second row shows foreground mask. Tracking output is shown in the third row, and the fourth row shows a rendering of the scene using the object-video stream associated with the person in Brown shirt. Notice that frames 266 and 308 (Row 4) also show a woman in a Blue top. This is an artifact of poor segmentation. Foreground detection erroneously merged blobs for the two individuals in frames 266 and 308. The blobs associated with the person in Brown shirt are shown in Violet.

Fig. 8 shows how blob coloring can improve scene awareness of an operator, while still preserving the privacy of individuals present in the scene. The Red blob shows a pedestrian who has crossed a virtual trip wire. Virtual trip wires, which are typically defined in pixel space, are routinely used in video surveillance systems to raise alarms.

6.2 Real Video Footage

Fig. 9 shows object-stream decomposition and subsequent selective rendering on real video footage. Fig. 9(e) renders pedestrians as colored blobs: multi-person blobs are shown in red and single person blobs are shown in blue. Tracker is unable to resolve the green blob in the top-left corner of the frame. Fig. 9(f) combines mean image estimated by observing 2000 frames and object-video streams for the two pedestrians in the bottom-right corner of the frame to render the scene showing only these two pedestrians. A closer look reveals ghosting artifacts in the rendered frame as the estimated mean frame is used to close the holes left by other pedestrians. Ghosting artifacts can be reduced by providing a reference background frame.

(a) (b) (c) (d) (e) (f)

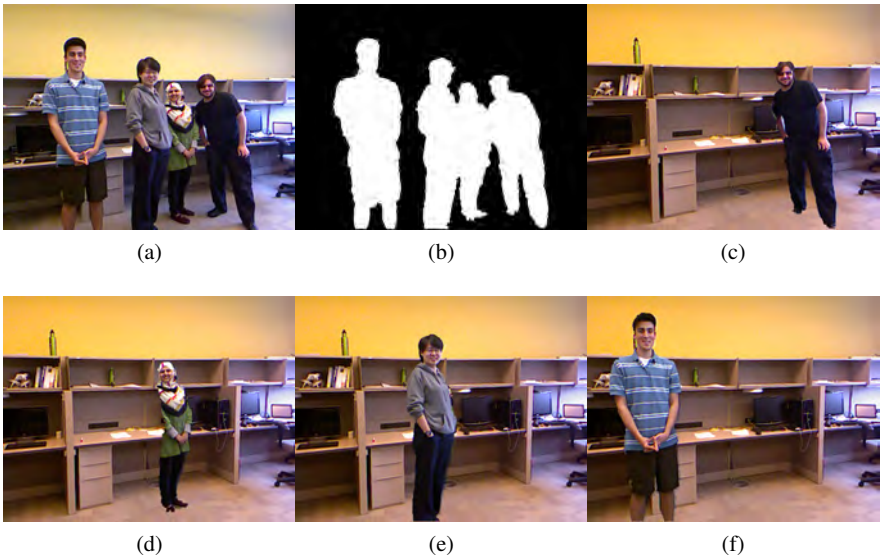
Fig. 9: Bootstrapping sequence from the Wallflower dataset [25]. (a) Raw frame, (b) mean image estimated using 2000 frames, (c) foreground mask, (d) pixel data for foreground objects, (e) showing all pedestrians as color blobs, and (f) re-imagining the scene with only two pedestrians.

6.3 Microsoft Kinect RGBD Sensor

Fig. 10 shows object-stream decomposition and subsequent selective rendering using Microsoft Kinect RGBD sensor. The captured video containing 3 individuals is decomposed into 3 object video streams, each containing only a single individual. In this case both color and depth information available through the Kinect sensor is used to construct the object video streams. Fig. 11 illustrates a situation where Kinect shines. The foreground mask shown in Fig. 11(a) shows a situation discussed in Sec. 4 where sometimes a single (connected) foreground region is associated to two or more individuals present in the scene. These situations are difficult to deal with in a general setting. Kinect sensor, however, can easily deal with these situations by relying upon the depth value associated with each pixel. In the example shown in Fig. 11, the foreground region (Fig. 11(b)) is decomposed into four individuals.

(a) (b) (c) (d)

Fig. 10: Using Microsoft Kinect RGBD images to construct object video stream. (a) Captured video and (b)-(d) object video streams constructed corresponding to the three individuals present in the scene.



(a) (b) (c)

(d) (e) (f)

Fig. 11: Using Microsoft Kinect RGBD images to construct object video stream. (a) Captured video, (b) foreground mask, and (c)-(f) object video streams constructed corresponding to the three individuals present in the scene.

6.4 Limitations

The work on privacy preserving video surveillance systems, including the work presented here, is focused on technical challenges related to obfuscating individuals present in the captured video stream. The underlying assumption is that the privacy of an individual is not violated if an operator is unable to see that person. While obfuscating individuals in captured video streams is a necessary first step towards realizing privacy preserving video surveillance system, this capability alone does *not* address the privacy issues surrounding pervasive video surveillance. This is not only true for the system presented here, but is also true for any system that attempts to hide the identity of an individual in the surveillance video.

Saini *et al*. [20] have developed privacy leakage models that attempt to quantify the loss of privacy due to video surveillance even when an individuals is never visually identified in any of the video streams. They cogently argue that privacy is compromised even in the presence of an obfuscation mechanisms that never fails. One a more practical note, however, it is worthwhile remembering that error tolerance for any obfuscation scheme is nearly zero. If the obfuscation scheme fails even for a single frame, the privacy of an individual is compromised.

7 Conclusions

We have proposed a novel framework for preserving privacy in video surveillance. Raw video data is decomposed into object-video streams. Such object-centric decomposition of the raw video presents new alternatives for upholding privacy policies and regulations in video surveillance. Object-specific privacy policies can be implemented. Object-video streams can be combined to recreate the original video, when warranted. Selective scene rendering, which focuses on a single aspect of the scene, is also supported.

The quality of object-based video decomposition is closely tied to the performance of low-level vision processing—poor segmentation leads to poor, or worse useless, video decompositions. Recent advances in background segmentation and pedestrian tracking suggest that the proposed approach is useful for scenes with low to medium crowd density. Pedestrian segmentation is still difficult in crowded scenes. It is conceivable that a privacy preserving scheme, such as ours, can be easily implemented in RGBD sensors similar to Microsoft Kinect. Many technical challenges, however, need to be addressed before such RGBD sensors can be used for video surveillance in general.

We are currently investigating encryption and access control mechanisms to develop secure rendering modules for video surveillance systems. These modules will combine object-video streams to present a mediated view of the scene to the operator. Such rendering modules are needed to gain the benefits of video surveillance technologies while preserving individual privacy. In closing we need to pay more attention to privacy implications of pervasive video surveillance. More work is needed to develop robust computer vision routines capable of striping identifiable information from surveillance footage without compromising the usefulness of the captured footage. Furthermore any privacy preserving video surveillance system must also take into account the privacy leakage channels inherent in pervasive video surveillance systems.

Acknowledgments

We thank Wei Shao and Mauricio Plaza-Villegas for their invaluable contributions to the implementation of the Penn Station simulator. We also thank Jordan Stadler for his work on constructing object-video streams using Microsoft Kinect sensor. This work is supported in part by the UOIT Startup Fund. We also acknowledge the NSERC Discovery Grant program.

References

[1] OpenKinect. http://openkinect.org/wiki/Main_Page (Last accessed on 28 May

2012).

[2] O D Arandjelovic and R Cipolla. Face Recognition from Video Using the Generic Shape-Illumination Manifold. In *Proc. European Conference on Computer Vision (ECCV06)*, volume 4, pages 27–40, Graz, Austria, May 2006.

[3] A M Berger. Privacy Mode for Acquisition Cameras and Camcorders. US patent 6,067,399 to Sony Corp., Patent and Trademark Office, 2000.

[4] L Bourdev and J Brandt. Robust object detection via soft cascade. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 05)*, volume 2, pages 236–243, San Diego, CA, June 2005.

[5] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., September 2008.

[6] A Chattopadhyay and T E Boult. PrivacyCam: a Privacy Preserving Camera Using uCLinux on the Blackfin DSP. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR07)*, pages 1–8, Minneapolis, MN, June 2007.

[7] Hwann-Tzong Chen, Horng-Horng Lin, and Tyng-Luh Liu. Multi-object tracking using dynamical graph matching. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR01)*, volume 2, pages 210–217, Hawaii, December 2001.

[8] F Dornaika and J Ahlberg. Fast and reliable active appearance model search for 3-d face tracking. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 34(4):1838–1853, 2004.

[9] Jianping Fan, Hangzai Luo, Mohand-Said Hacid, and Elisa Bertino. A novel approach for privacy-preserving video sharing. In *Proc. 14th ACM international conference on Information and knowledge management (CIKM05)*, pages 609–616, New York, NY, USA, November 2005. ACM.

[10] K Kim, T H Chalidabhongse, D Harwood, and L Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11:167–256, March 2005.

[11] Malcolm Kirkup and Marylyn Carrigan. Video surveillance research in retailing: ethical issues. *International Journal of Retail & Distribution Management*, 28(11):470–480, 2000.

[12] Timothy Moser, Dwayne Nelson, Richard Williams, and Rick Rowe. Casino Patron Tracking and Information Use. US patent WO/2008/067212, June 2008.

[13] Marcus Nieto, Kimberly Johnston-Dodds, and Charlene Wear Simmons. *Public and Private Applications of Video Surveillance and Biometric Technologies*. 2002.

[14] Clive Norris, Mike McCahill, and David Wood. The growth of CCTV: a global perspective on the international diffusion of video surveillance in publicly accessible space. volume 2, pages 110–135. 2004.

[15] J M Ogden, E H Adelson, J R Bergen, and P J Burt. Pyramid-based computer graphics. Technical Report RCA Engineer 30-5, RCA Corporation, September 1985.

[16] F Z Qureshi and D Terzopoulos. Smart Camera Networks in Virtual Reality. *Proceedings of the IEEE (Special Issue on Smart Cameras)*, 96(10):1640–1656, October 2008.

[17] Faisal Z Qureshi. Object-Video Streams for Preserving Privacy in Video Surveillance. In *Proc. 6th International Conference on Advanced Video and Signal Based Surveillance (AVSS 09)*, pages 1–8, Genovo, Italy, September 2009.

[18] Mukesh Saini, Pradeep K Atrey, Sharad Mehrotra, Sabu Emmanuel, and Mohan Kankanhalli. Privacy Modeling for Video Data Publication. In *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, IEEE International Conference on Multimedia and Expo, pages 60–65, Singapore, July 2010.

[19] Mukesh Saini, Pradeep K. Atrey, Sharad Mehrotra, and Mohan Kankanhalli. Adaptive Transformation for Robust Privacy Protection in Video Surveillance. *Advances in Multimedia*, 2012:1–14, 2012.

[20] Mukesh Saini, PradeepK. Atrey, Sharad Mehrotra, and Mohan Kankanhalli. W3-privacy: understanding what, when, and where inference channels in multi-camera surveillance video. *Multimedia Tools and Applications*, pages 1–24, 2012.

[21] J Schiff, M Meingast, D K Mulligan, S Sastry, and K Goldberg. Respectful cameras: Detecting visual markers in real-time to address privacy concerns. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pages 971–978, San Diego, CA, November 2007.

[22] A Senior, S Pankanti, A Hampapur, L Brown, Ying-Li Tian, A Ekin, J Connell, Chiao Fe Shu, and M Lu. Enabling video privacy through computer vision. *IEEE Transactions on Security and Privacy*, 3(3):50–57, 2005.

[23] Andrew Senior, Sharath Pankanti, Arun Hampapur, Lisa Brown, Ying-li Tian, and Ahmet Ekin. Blinkering Surveillance : Enabling Video Privacy Through Computer Vision. Technical report, IBM, NY, 2003.

[24] W Shao and D Terzopoulos. Autonomous Pedestrians. *Graphical Models*, 69(5-6):246–274, 2007.

[25] Kentaro Toyama, John Krumm, Barry Brumitt, and Brian Meyers. Wallflower: Principles and Practice of Background Maintenance. In *Proc. IEEE International Conference on Computer Vision (ICCV99)*, volume 1, pages 255–261, Kerkyra, Greece, September 1999.

[26] Paul Viola and Michael Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR01)*, pages 1–8, Hawai, December 2001.

[27] Jyoji Wada, Koji Wakiyama, Haruo Kogane, and Noboru Takada. Monitor Camera System and Method of Displaying Pictures from Monitor Camera Thereof. European patent EP 1 081 955 A3 to Matsushita Electric Industrial, European Patent Office, 2001.

[28] Bo Wu and Ram Nevatia. Detection of Multiple, Partially Occluded Humans in a Single Image by Bayesian Combination of Edgelet Part Detectors. In *Proc.*

Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, pages 90–97, Beijing, China, October 2005.