Object-Video Streams for Preserving Privacy in Video Surveillance

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

***Abstract*—This paper presents a framework for preserving privacy in video surveillance. Raw video is decomposed into a background and one or more object-video streams. Object-video streams can be combined to render the scene in a variety of ways: 1) The original video can be reconstructed from object-video streams without any data loss; 2) individuals in the scene can be represented as blobs, obscuring their identities; 3) 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; 4) the scene can be partially rendered, i.e., revealing the identities of *some* individuals, while preserving the anonymity of others. We evaluate our approach in a virtual train station environment populated by autonomous, lifelike virtual pedestrians.**

***Keywords*-video surveillance; privacy; object-video streams;**

I. Introduction

Video surveillance is ubiquitous. Many cities around the world are increasingly relying on video surveillance for crime prevention and community safety. Video footage captured through surveillance cameras 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. In a recent article that appeared in *Times Online UK*, K. Burgess wrote about the extent of Britain's surveillance society. She counted 281 cameras along her 3.1 mile long route to work. Of these 173 cameras were installed by private operators.

The panoptic effect of pervasive video surveillance raises many questions. Who is collecting information about us? How this information is being used? What information is being collected? Who has access to this information? 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. 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.

This paper presents a framework for privacy preserving video surveillance systems. Camera video is processed to construct *object-video* streams. Object-video streams can be combined to view the scene in a number of different ways. For example, individuals can be represented as color blobs. Operators can thus see scene activity without knowing the identities of the people present in the scene. Blob colors can convey subtle information about the scene to the operator. Object-video streams can be combined to recreate the original video. Object-video streams can also be combined to make only certain individuals visible.

We embrace the *Virtual Vision* paradigm, exploiting visually and behaviorally realistic virtual environments to develop and empirically evaluate our video surveillance framework [1]. We employ a virtual train station environment populated by autonomous lifelike virtual pedestrians that is described in [2]. The vision pipeline for our prototype video surveillance system matches the performance of the vision pipeline (for real video) presented in [3]. Therefore, the obtained results are legitimate and valuable. We describe vision pipeline in Sec. III. We also show object-stream construction and selective rendering using real video footage in Fig. 8.

II. 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 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 [4]–[8]. 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 [9], [10]. 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 [11]. 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 [12]. [13] patented a system capable of obscuring a privacy region in a pan-tilt-zoom camera. [14] 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* [10]. 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.

III. Vision Pipeline

We setup a video surveillance system comprising static wide field-of-view cameras in our virtual vision simulator. The performance of the proposed surveillance system is ultimately tied to the capabilities of the vision pipeline that is responsible for analyzing raw video data. Our vision pipeline, which operates on synthetic video, consists of established vision algorithms, and it can be adapted for real video with minimal effort (Fig. 1).[1] Vision algorithms operate solely upon the synthetic video captured by virtual cameras.

A. 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 [15]. We use the implementation of codebook method for background learning provided in the Open Computer Vision Library (OpenCV) [16]. 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 [3]. 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.

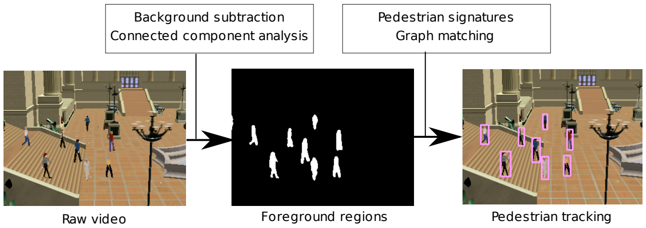


Figure 1: Vision pipeline: We have adapted well-understood computer vision algorithms for our purposes. The vision routines operate solely upon synthetic video captured by virtual cameras. Background subtraction is used to identify foreground pixels. Pedestrians signatures that encode pedestrian color distribution in HSV space are matched in successive frames to perform tracking.

B. 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 have developed an appearance-based pedestrian tracker that mimics the performance of a state-of-the-art pedestrian tracker. In particular, tracking can fail due occlusions, poor segmentation, bad lighting, or crowding. Pedestrian signatures encode pedestrian color distribution as a 3D histograms in Hue-Saturation-Value (HSV) color-space. We have empirically selected 32 bins along the first two (Hue-Saturation) dimensions and 16 bins along the last (Value) dimension. Tracking is performed by setting up a bipartite graph matching problem as suggested in [3]. The optimal solution to the matching problem resolves pedestrian identities across multiple frames. We refer the reader to [3] 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.

[1] E.g., the vision pipeline developed for synthetic video is used to construct object-video streams from real video footage in Fig. 8.

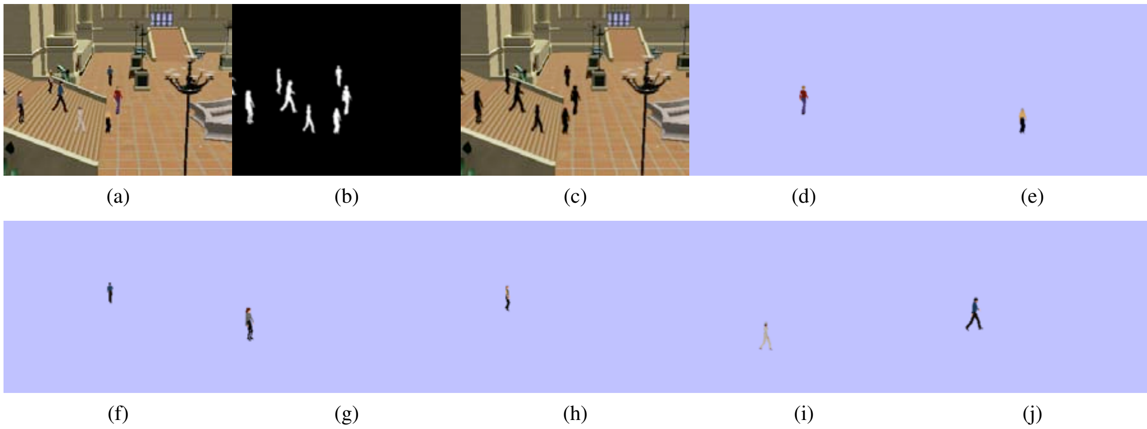


Figure 3: Decomposing video into a background component and 7 foreground components. Each foreground component encodes visual data for a particular pedestrian. (a) Raw video. (b) Foreground mask. (c) Background image containing holes. (d)-(j) RGBA frames containing color data for 7 pedestrians visible in the frame.

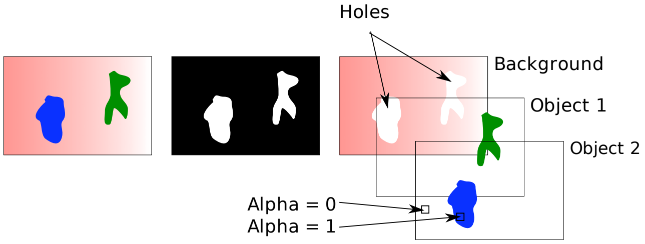


Figure 2: Cleaned up foreground mask decomposes a video frame into a background component and two foreground components. Pedestrian to blob mapping information maintained by the pedestrian tracker links each foreground component to one (or more) pedestrians.

IV. 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:

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

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.

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

$A(C_i)$ denotes the area enclosed by blob C_i.

2) Construct an RGBA color image F_t^i.

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

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. 2 and 3, 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. 4. 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.

V. 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 [17]. Laplacian pyramid blending is also used to fill the holes in the rendered scene by using the stored average background image F^B.

- 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 other persons present in the scene.

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.

VI. 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 simplify pedestrian identity management across multiple cameras. Decomposing raw video into object-video streams does not require the camera network to be calibrated.

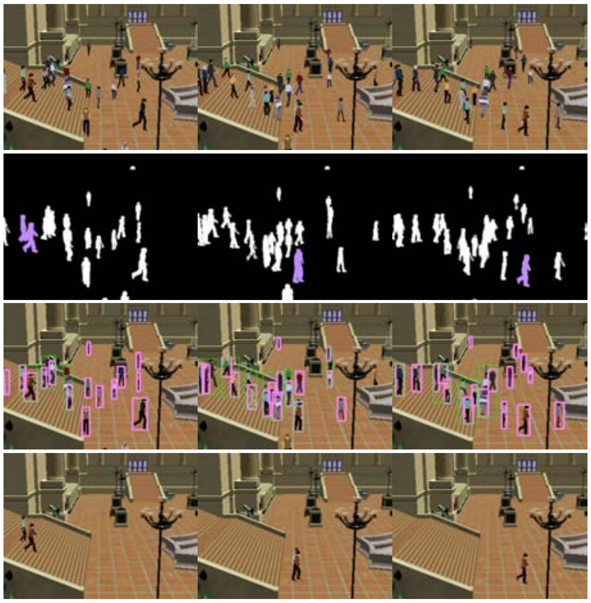

Frame 266 Frame 308 Frame 325

Figure 6: 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.

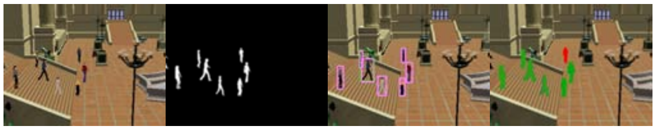
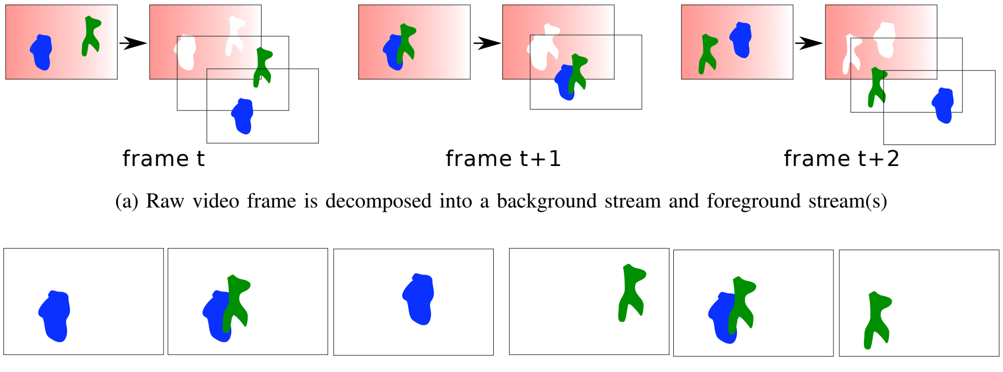

(a) (b) (c) (d)

Figure 7: 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.

We show different rendering possibilities in Fig. 5. Fig. 5(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. 5(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. 5(e) is showing the 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. 6 shows selective rendering. The top row contains

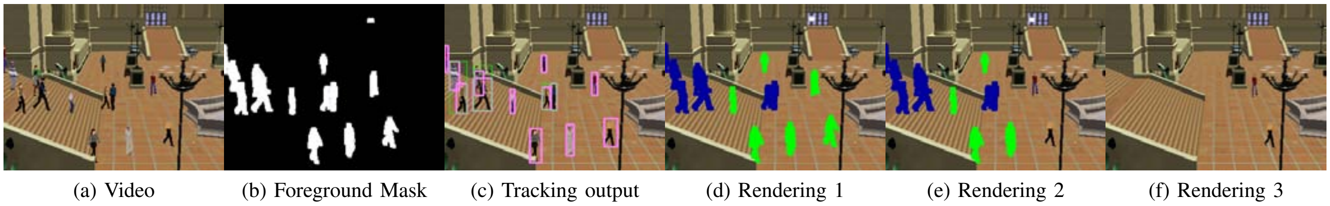


(a) Raw video frame is decomposed into a background stream and foreground stream(s)

(b) Object-video stream for Blue object

(c) Object-video stream for Green object

Figure 4: Constructing object-video streams.



(a) Video (b) Foreground Mask (c) Tracking output (d) Rendering 1 (e) Rendering 2 (f) Rendering 3

Figure 5: Decomposition into object-video streams presents new possibilities to view the scene.

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. 7 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.

Fig. 8 shows object-stream decomposition and subsequent selective rendering on real video footage. Fig. 8(e) renders pedestrians as colored blobs: multi-person blobs are shown in red and single person blobs are show in blue. Tracker is unable to resolve the green blob in the top-left corner of the frame. Fig. 8(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.

VII. 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 difficult in crowded scenes.

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.

Acknowledgments

We thank Wei Shao and Mauricio Plaza-Villegas for their invaluable contributions to the implementation of the Penn

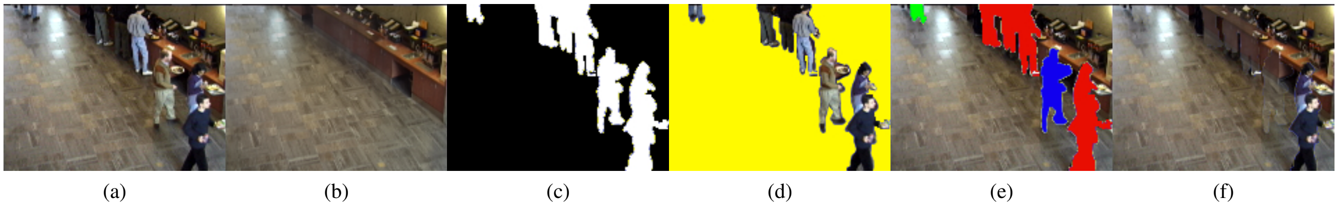


Figure 8: Bootstrapping sequence from the Wallflower dataset [18]. (a) Raw frame, (b) mean image estimated using 2000 frames, (c) foreground mask, (d) pixel data for foreground objects, (e) showing all pedestrians as color blobs, and (f) re-imagining the scene with only two pedestrians.

Station simulator. We thank reviewers for their valuable comments and suggestions. This work is supported in part by the UOIT Startup Fund. We also acknowledge the NSERC Discovery Grant program.

References

[1] F. Z. Qureshi and D. Terzopoulos, "Smart camera networks in virtual reality," *Proceedings of the IEEE (Special Issue on Smart Cameras)*, vol. 96, no. 10, pp. 1640–1656, October 2008.

[2] W. Shao and D. Terzopoulos, "Autonomous pedestrians," *Graphical Models*, vol. 69, no. 5-6, pp. 246–274, September/November 2007.

[3] H.-T. Chen, H.-H. Lin, and T.-L. Liu, "Multi-object tracking using dynamical graph matching," in *Proc. of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR01)*, vol. 2, Hawaii, December 2001, pp. 210–217.

[4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR01)*, Hawai, December 2001, pp. 1–8.

[5] 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*, vol. 34, no. 4, pp. 1838–1853, 2004.

[6] L. Bourdev and J. Brandt, "Robust object detection via soft cascade," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 05)*, vol. 2, San Diego, CA, June 2005, pp. 236–243.

[7] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors," in *Proc. of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Beijing, China: IEEE Computer Society, October 2005, pp. 90–97.

[8] O. Arandjelovic and R. Cipolla, "Face recognition from video using the generic shape-illumination manifold," in *Proc. European Conference on Computer Vision (ECCV06)*, vol. 4, Graz, Austria, May 2006, pp. 27–40.

[9] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Y.-L. Tian, A. Ekin, J. Connell, C. F. Shu, and M. Lu, "Enabling video privacy through computer vision," *IEEE Transactions on Security and Privacy*, vol. 3, no. 3, pp. 50–57, May-June 2005.

[10] A. Chattopadhyay and T. Boult, "Privacycam: a privacy preserving camera using uclinux on the blackfin dsp," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR07)*, Minneapolis, MN, June 2007, pp. 1–8.

[11] J. Schiff, M. Meingast, D. 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)*, San Diego, CA, November 2007, pp. 971–978.

[12] A. M. Berger, "Privacy mode for acquisition cameras and camcorders," US patent 6,067,399 to Sony Corp., Patent and Trademark Office, 2000.

[13] J. Wada, K. Wakiyama, H. Kogane, and N. 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.

[14] J. Fan, H. Luo, M.-S. Hacid, and E. Bertino, "A novel approach for privacy-preserving video sharing," in *Proc. 14th ACM international conference on Information and knowledge management (CIKM05)*. New York, NY, USA: ACM, November 2005, pp. 609–616.

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

[16] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., September 2008.

[17] J. Ogden, E. Adelson, J. R. Bergen, and P. Burt, "Pyramid-based computer graphics," RCA Corporation, Tech. Rep. RCA Engineer 30-5, September 1985.

[18] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. of the IEEE International Conference on Computer Vision (ICCV99)*, vol. 1, Kerkyra, Greece, September 1999, pp. 255–261.