2011 Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics

The De-Identification Camera

Mrityunjay, P. J. Narayanan
International Institue of Information Technology
Hyderabad
mrityunjay@research.iiit.ac.in, pjn@iiit.ac.in

***Abstract*—Visual surveillance is increasingly prevalent today but the privacy issues of individuals involved in surveillance videos have not been dealt with adequately so far. In most cases, if not all, the chief purpose of placing a camera can be served without knowing the identity of the individuals involved unless the activity is of some predetermined kind. One needs to transform these videos to protect identity of individuals involved possibly at the source camera itself. In this paper we present the De-Identification Camera, which is a scalable, low cost and real-time solution to the privacy protection issues. Our main contribution lies in proposing a privacy protection architecture which transforms the video at the camera level itself, We also present and implement a de-identification pipeline which is suitable for real-time implementation. We implemented the system on a Texas Instrument OMAP4 based embedded platform and was able to de-identify videos in real time, transforming the video on the camera itself ensures protection from various attacks. We also address issues like data utility of surveillance videos by making this solution customizable. We propose that such systems can replace the traditional surveillance cameras in the future by providing all the surveillance and privacy protection solutions on hardware, probably with few performance upgrades.**

Keywords- De-identification, privacy protection, embedded vision.

I. Introduction

Video surveillance is a rapidly growing necessity today. The decrease in the costs of hardware and bandwidth has made video surveillance system cheaper and easier to install widely. Cameras may be watching us everywhere we go, be it a public place, a private enterprise, or public transportation mode. People are monitored by cameras and the data is recorded continuously at different servers. There may also be evidence in terms of the crime rates that large scale surveillance has positive impact [1]. Observing people in video is also important to study customer behavior, traffic analysis, crowd dynamics, etc. This data can be used to train computer vision and machine learning algorithms to make automatic analysis effective and efficient [2], [3].

Video surveillance is causing concerns on the issue of the violation of privacy of individuals involved. Data captured by services like Google street view [4], Bing Street-view Map, Map-Jack etc.are available on the Internet for general public viewing. Lately these concerns are highly raised in several countries. The privacy could be violated by a motivated individual watching the images or videos, or even by automated agents powered by state-of-the-art computer vision techniques. The ability of algorithms to detect faces or humans and to track them is considerable today [5]–[8].

We need to ensure that the surveillance videos cannot be put to undesirable uses and that privacy of individuals is protected. An effective system should hide all aspects relating to the identity of the individual while preserving sufficient information to comprehend what action has been performed. De-identification is a process of removal of particular aspects of person's identification from images and videos [10], [11] by manipulating them. Several approaches are possible to remove identifying information from videos and images. Preserving other information while hiding the identity may not be easy to achieve as there is a natural trade-off between the two.

The de-identification transformation should also be immune to human and computer vision. It is always safe to hide all the information which leads to recognition of the person by, say, object impainting but this also hides most of the information on the sorts of human activities going on in that space, which can be useful for various other purposes. It is best if videos and images are de-identified right at capture time by the camera with embedded processing and save a clean copy of the video protected by strong encryption and authentication technique which could be used in case of emergency only by authorized personnel. In this work we present a system which does the de-identification right at the camera level. Our system can also be customized to facilitate different levels of data utility by providing different de-identification transformations.

II. Related Work

In the past few years many solutions in the area of privacy protection were proposed, these systems address the problem of protecting privacy of individuals in services like Google Street view, web videos and other form of visual data available through the internet. Some approaches follow the practices in traditional print media, these image distortion methods modifies the region of the image where a person is present using image filters such as blurring using Gaussian filter [4] or pixelation. These methods are easy to use but doesn't guarantee privacy or data utility [3] as they are susceptible to reverse engineering attacks. These systems can be very useful to deploy where sophisticated privacy protection is not a big concern [12]. An alternative approach in the video surveillance domain which provides some user control on the amount of distortion applied in the image [13], This method applies background subtraction in the video feed and followed by that they scramble the coefficients used to encode the area in a Motion JPEG compressed video sequence in region of interest

978-0-7695-4599-8/11 $26.00 © 2011 IEEE
DOI 10.1109/NCVPRIPG.2011.48

of the image. The magnitude of change is controlled by the user.

Face De-identification [11] is another method of privacy protection. Gross et al.explained the effectiveness of blurring the faces in protecting a person's identity by testing the system against face recognition system. This method has limited usage since people use more information than just faces to identify people in images and videos [14].

Person De-identification in videos [10] targets privacy protection issues in surveillance videos, instead of just faces, they target the complete human body in spatio-temporal domain and mask it using three different type of blur functions in a video. Chinomi et al. [15] replace persons with the background color pixels or by solid colored bounding boxes. Such a method is useful in extreme situations when the surveillance agency is not interested in monitoring people at all.

Most of the earlier work in this field is focused on providing off line solution to privacy protection, i.e. the videos are stored and processed offline before publishing on the web. A real time privacy protection system, Respectful Cameras was proposed by Schiff et al. [16], this system requires the users opting for privacy protection to wear colored markers which are tracked by the system and real time and face of users wearing colored markers is obscured from the video. TrustCAM [17] is another privacy protection system which implements the system on a prototype, although their work is more focused towards using data encryption to hide an individual's identity instead of using computer vision. PrivacyCam [18] is a system implemented on a DSP architecture using computer vision techniques but they transform only the face of a person to protect his or her identity by scrambling the coefficients used for JPEG image encoding. The majority of the existing work doesn't focus on real time implementation of a privacy protection but rather concentrate on providing a theoretical or offline solution. We see our work as a step towards taking these systems to real time systems which can be actually deployed and used in surveillance systems.

III. The De-Identification Camera

In this paper we propose and present a solution to the real-time privacy protection problem. Our system, the De-identification Camera, is prototyped using a low cost embedded platform. In this section, we propose few features that every privacy protection system should ideally have. We also discuss the goals that we have achieved in our work, underlying assumptions of our work and description of the de-identification hardware and software used in our proposed system.

A. Goals of our system

A privacy protection system cannot defeat the purpose of placing the camera by hiding everything. At the same time the system should be robust against possible software attacks, scalable and efficient for surveillance applications. We target following design goals for our system and try to achieve them in our prototype.

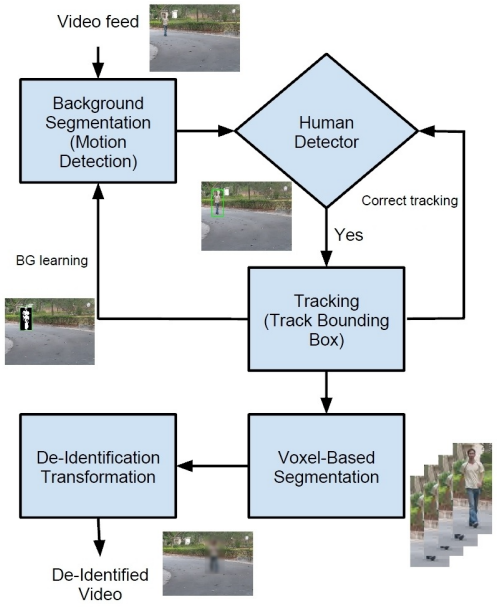

Fig. 1. The De-identification pipeline

1) De-identify at source: Processing the videos at the camera hardware itself ensures minimal or no attacks due to sniffing of information while data transfer. The method should be strong enough so that unintended information cannot be obtained even if the channel is compromised. Physical access to the camera should be constrained for the security of the system.
2) Scalable Solution: The system should be scalable in terms of hardware and software capabilities. Hardware capabilities can be extended by using widely available embedded platforms, GPUs, multi-core systems. We use Texas Instruments OMAP based hardware platform which is stable and capable. Scalability in terms of software architecture can be achieved by using independent modules in the pipeline which can be replaced by better and more efficient modules as and when there is a requirement or availability while maintaining the rest of the pipeline intact.
3) Customized De-Identification: Smart cameras used in surveillance systems can be deployed at various locations. In an airport, observing a person's activity is far more important than protecting his privacy. The system can be used as a street camera where knowing an individual's identity is not the prime goal. Our system provides customization option for de-identifying the videos, the user can choose a trade-off between privacy or data utility by setting the level of de-identification required by the system.
4) Real time performance: Any surveillance system which is used for real time monitoring of videos should process the data in real time for being suitable for deployment.

The de-identification pipeline that we use in our system gives a near real-time performance on our current hardware, better performance can be achieved by using more powerful hardware.

B. System Description

An overview of our De-identification process is outlined in Figure 2. Our pipeline is influenced by the one used by Agarwal et al. [10] but we have optimized the pipeline to deliver real time performance on the embedded hardware. To evaluate the performance of the pipeline we implement it on an OMAP4 based hardware. In the following sections, we first describe the de-identification pipeline used in our system followed by the description of the hardware and software architecture.

1) De-identification Pipeline: The de-identification pipeline that we use in our system is comprised of five modules: Background subtraciton, Detection, Tracking, Segmentation and transformation for de-identification.

1) Background Subtraction: In the first step of the pipeline we segment out the moving foreground objects from a static learned background. We follow the Extended Gaussian mixture model (GMM) [19] approach for background subtraction (BGS). The foreground objects detected in this module of the pipeline serve as an input to the detection module.
2) Person Detection: Once we have foreground objects in the scene we apply HOG based person detector [5]. The detector is only applied on the region with foreground objects in it and the image is downsized to 0.4 of its original size to increase detector performance. The detection time is reduced from 1800ms to 90ms as a result of image down sampling for detecting one person in a 640x480 image on our system. We use OpenCV implementation of HOG based person detector in this module. The detector returns a bounding box which fits the person with-in, we also increase the size of this bounding box to avoid the case when the detected bounding box fails to enclose complete human body. The bounding box is passed to the tracking module and is only called periodically based on the movement in the scene that is being monitored. We call the detector more often if scene has more noise which can adversely affect the performance of the tracking module. For our experiments we call the detector every 5 frames.
3) Track Person: The bounding box returned by the detector module is then tracked across time to track the person's location in the video, since tracking is much faster than detection this improves the overall performance of the system. We use the continuously adaptive mean-shift tracking algorithm [7] for tracking the bounding box in the video. The reliability of tracking is taken care by tuning the time period at which detector is called within the tracking module. Even if the person stops moving in the scene the detector will be able to refresh tracker's memory to detect a stationary person.

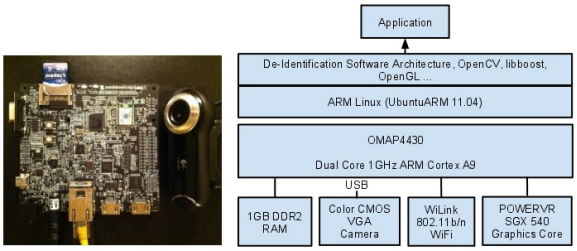

Fig. 2. System architecture, OMAP4 based hardware, architecture of the system

4) Segmentation: The tracking module gives a bounding box containing a person in it, these bounding boxes are stacked across time to generate a video tube of the person. Instead of performing costly graph cut based segmentations to extract fine silhouette information about the person we choose to have the video tube as big as the size of the bounding box. The de-identification transformation is only applied in the segmented video tube.
5) De-Identification Transformation: After the segmentation of the video tube, we apply the de-identification transformation to bounding box region of the image. We use two types of de-identification transformation to obscure the person's identity: 1) Gaussian Blur of the pixels inside the bounding box, and 2) Binarizing the intensity values of the bounding box image. Different variations of these transformations are used in our method and more such transformations can be plugged into the pipeline to get better and variable results.

2) Hardware and Software Architecture: The hardware prototype that we have used in our De-identification Camera system is a commercially available development platform from Texas Instrument. The system is based on TI's OMAP4430 application processor, which is a 1GHz Dual Core ARM Cortex A9 processor. The system has a POWERVR SGX540 graphics core along with 1GB of DDR2 RAM. We connect a color VGA CMOS sensor (Logitech QuickCam Pro 9600) via USB to get video input. The system is connected via a WiLink 802.11n wireless link which is used to control it and view the de-identified video on a host computer.

On the software side, we use ARM Linux (UbuntuARM) and libraries like OpenCV 2.2, libboost, Qt, OpenGL to implement our system. Figure 2. gives an overview of the hardware and software architecture of De-identification camera system.

IV. Results and Discussion

Figure 3. shows the set of results we obtained as a result of applying our method on a video feed of VGA resolution. Currently in our experiments we are assuming that the camera is stationary and only one person appears in the scene at a time. Except HOG based person detector every other module is called for each frame. Detector is called every fifth frame

Module	Run-time (ms)
BG Subtraction	10
Human Detector (used periodically)	210
Camshift Tracker	10
De-id Transform	15
Other (pre-processing, etc)	10
Average run-time per frame	90-105

TABLE I
PERFORMANCE ANALYSIS OF THE METHOD.

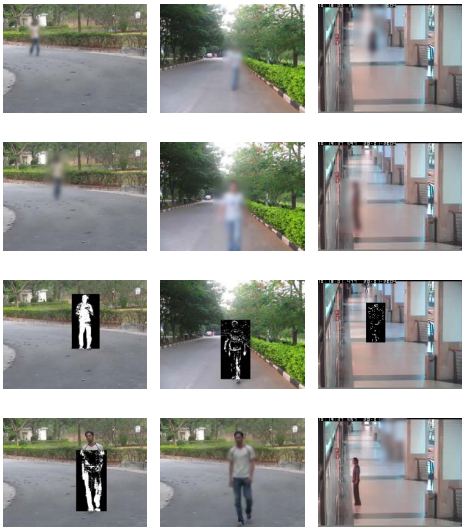

Fig. 3. De-identification results from our system. Row 1. shows the de-id results after applying Gaussian Blur with window size 21, in row 2. Gaussian Blur with window size 41, in row 3. we binarize the image inside the bounding box and row 4. shows the case when the method fails to cover up the person's identity

in a 25fps video stream, it takes approx 210ms to process one down-sampled frame. Background subtraction (BGS), tracking, de-identification transform takes another 35-50ms to process one frame. Other operations like image pre-processing, etc.take 10-15ms to complete the whole pipeline. On average the complete pipeline takes 90-105ms for one frame, which turns out to be a speed of 10-11fps for the whole system. The run time for individual modules is shown in Table 1. The inconsistency in the detection and tracking modules results in a miss sometimes and a part of the person's body is visible in one of the frames as shown in the fourth row of fig 3. The hit rate for de-identifying a person is about 0.97 and most of miss occurs when the person is too close to the camera or not visible completely. We are able to achieve near real time performance using our method on the specified hardware. Better speedup and performance can be gained by improving detection and tracking algorithms for the specific purpose of de-identification.

V. CONCLUSION

In this paper, we addressed the issue of real-time privacy protection of an individual in videos. We discussed the importance of having a strong privacy protection policy at the camera level itself to protect the system against leakage and reverse engineering attacks. We presented our system, the de-identification camera which implements these policies on an embedded hardware platform and provides de-identified video with real-time performance. In addition to being real-time, the policy is also customizable so that the system can be tweaked to output de-identified video satisfying various data utility needs. The proposed system is easily scalable using the advancements in hardware technology and performance improvement of detection and tracking modules.

REFERENCES

[1] A. W. Senior, S. Pankanti, A. Hampapur, L. M. G. Brown, Y. li Tian, A. Ekin, J. H. Connell, C.-F. Shu, and M. Lu, "Enabling video privacy through computer vision," *IEEE Security & Privacy*, vol. 3, no. 3, pp. 50–57, 2005.

[2] G. T. Duncan, S. A. Keller-mcnulty, and S. L. Stokes, "Disclosure risk vs. data utility: The r-u confidentiality map," Chance, Tech. Rep., 2001.

[3] G. Loukides and J. Shao, "Data utility and privacy protection trade-off in k-anonymisation," in *PAIS*, 2008, pp. 36–45.

[4] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. S. Ogale, L. Vincent, and J. Weaver, "Google street view: Capturing the world at street level," *IEEE Computer*, vol. 43, no. 6, pp. 32–38, 2010.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR (1)*, 2005, pp. 886–893.

[6] P. A. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.

[7] B. Kwolek, "Camshift-based tracking in joint color-spatial spaces," in *CAIP*, 2005, pp. 693–700.

[8] H. A. Aboalsamh, "Human face recognition using eigen and fisher faces," *Egyptian Computer Science Journal*, vol. 31, no. 1, 2009.

[9] B. Boufama and M. A. Ali, "Tracking multiple people in the context of video surveillance," in *ICIAR*, 2007, pp. 581–592.

[10] P. Agrawal and P. J. Narayanan, "Person de-identification in videos," in *ACCV (3)*, 2009, pp. 266–276.

[11] R. Gross, E. Airoldi, B. Malin, and L. Sweeney, "Integrating utility into face de-identification," in *Privacy Enhancing Technologies*, 2005, pp. 227–242.

[12] C. Neustaedter, S. Greenberg, and M. Boyle, "Blur filtration fails to preserve privacy for home-based video conferencing," *ACM Trans. Comput.-Hum. Interact.*, vol. 13, no. 1, pp. 1–36, 2006.

[13] F. Dufaux and T. Ebrahimi, "Scrambling for privacy protection in video surveillance systems," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 18, no. 8, 2008.

[14] M. Nishiyama, H. Takeshima, J. Shotton, T. Kozakaya, and O. Yamaguchi, "Facial deblur inference to improve recognition of blurred ffaces," in *CVPR*, 2009, pp. 1115–1122.

[15] K. Chinomi, N. Nitta, Y. Ito, and N. Babaguchi, "Prisurv: Privacy protected video surveillance system using adaptive visual abstraction," in *MMM*, 2008, pp. 144–154.

[16] J. Schiff, M. Meingast, D. K. Mulligan, S. Sastry, and K. Y. Goldberg, "Respectful cameras: detecting visual markers in real-time to address privacy concerns," in *IROS*, 2007, pp. 971–978.

[17] T. Winkler and B. Rinner, "Trustcam: Security and privacy-protection for an embedded smart camera based on trusted computing," in *Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, ser. AVSS '10, 2010.

[18] A. Chattopadhyay and T. E. Boult, "Privacycam: a privacy preserving camera using uclinux on the blackfin dsp," in *CVPR*, 2007.

[19] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," 2004.