# Privacy Preservation of CCTV Image Stream

Gurnoor Singh
COMP 7860
Department of Computer Science
University of Manitoba, Canada
kainthg1@myumanitoba.ca

Manmohit Singh
COMP 7860
Department of Computer Science
University of Manitoba, Canada
singhm30@myumanitoba.ca

***Abstract*—The recent advances in CCTV surveillance technology allows a person to efficiently monitor and capture majority of the daily activities performed by people in the video. It presents a significant threat to the individual privacy as this data, if misused, may expose highly sensitive and private parameters such as the attitude, location and behavior along with the evident identity of a person. We present a system to be used with CCTV image data stream that ensures data utility and performs efficient video analysis without violating privacy. This system is intended to focus on specific use cases that require identification of a general trend among a crowd rather than individual monitoring. We design an algorithm that uses a mask to generate a privacy protecting layer over the original image data stream, ensuring appropriate distortion of the personal identity. We further propose two approaches for the image distortion, "Contours" and "Blob" depending upon the vital distortion necessities for a given scenario. We validate the system by applying this algorithm to a real-time image data stream captured by a CCTV camera in public setting. We are able to show that this system is highly effective in protecting the individual privacy in the aforementioned use cases without affecting data utility.**

***Keywords—component; de-identification; image stream; masking; computer vision (key words)***

## I. INTRODUCTION

Due to the rising need for smart buildings, extensive research is being done in the field of building design. One of the key components of this research is to design optimal emergency evacuation models. Different models have been proposed over the years to make the process as efficient as possible. Bensilum and Purser [1] developed 'GridFlow' to represent individual occupants in building spaces on a grid network, moving to exit points through escape routes using an x, y coordinate, and distance map method. Ma et el. [2] performed experiments in Shanghai World Financial Center, involving a mass evacuation process which was captured by CCTV video cameras. The data captured was extracted out manually for analysis of movement characteristics such as mean speed and the time needed for evacuation. We can easily infer that the data collected in such experiments study the general trend of movement and the trajectories and does not require analysis at an individual level.

The raw data, even when collected for the purpose of research, violate individual privacy as it can expose sensitive attributes such as location data. Beresford and. Stajano [3] highlight that on inspecting the history of all the past movements of an individual, recorded every second with sub-meter accuracy, can be of significant qualitative value and hence has the potential to be misused by miscreants.

**Our Contribution :** In order to solve this conundrum, we propose a privacy-preserving system based on algorithms designed to de-identify people by distorting there identity. We introduce an edge over blurring and smoothening the facial portion by completely fussing the overall image of a person with a complete noise, using either of "Contours" or "Blob" proposed approaches. As a person could be identified earlier as the facial attributes even after applying various adhoc methods could lead to further privacy breach. Our method allows for a privacy-protecting stream output which can be used without a significant drop in data utility. Furthermore, our model doesn't require any image stream pre-processing at all. It can simply be applied to any image stream irrespective of the video resolution, frame rate or size. In this context, we describe two approaches: one preferring the privacy preservation (Blob) and other preferring data utility (Contours). Finally, we apply this system to a stream of image data from a CCTV camera placed in a public setting.

*Table 1: Relation between various models*

| Risk-Utility Tradeoff Evaluations | Utility preservation | Privacy Preservation | | | | |
|---|---|---|---|---|---|---|
| | | Background subtraction | Adhoc methods | | | Foreground Masking |
| | | | Blurring / Pixelization | Bar/T Mask | Approximation | |
| Neustaedter et el. [5] | | | | * | | |
| Newton et el. [8] | | | * | | | |
| Gross et el. [7] | * | | | | * | |
| Zhang et el. [22] | | * | | | | |
| Our Contribution | * | | | * | | * |

## II. BACKGROUND INFORMATION

In this section, we have summarized and discussed some essential preliminaries tools that were required to develop this model.

**Microsoft Visual Studio** is an integrated development environment (IDE) from Microsoft as defined by Wikipedia [24]. It is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code. It includes code editor as well as code refactor. It holds the capability to debug the code at the Machine level along with at the source level. Since our model is based on C++ language we use Visual C++ libraries to program in the visual studio. Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring [24]. The integrated debugger does the job of both source-level debugger and a machine-level debugger. Along with these other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer [24]. It holds the capability of adding additional plug-ins as we added the Visual C++ libraries, which results in enhanced productivity. This software can support 36 different programming languages by just installing their respective library files.

**Open CV** is a library of programming functions that mainly aim at the real-time computer vision [26]. Originally developed by Intel, it was later supported by Willow Garage and is now maintained by Itseez [27]. The library is cross-platform and free for use under the open-source BSD license [25]. Open CV has its application in various productive fields like object and face identification, motion understanding, segmentation, recognition, structuring from motion along with tracking motion and depth perception. Open CV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE also, though all of the new developments and algorithms in Open CV are now developed in the C++ interface [25].

## III. RELATED WORK

### *A. Privacy Preservation of Events*

From past few decades, work is being done on preserving the privacy of visual data, such as surveillance CCTV camera recordings. To avoid the misuse of such data, many techniques and methods were proposed: these include many ad-hoc methods like Blurring and Pixilation for distorting the facial characteristics, so that these images could not be re-identified using the Test data as proposed by Newton et el. [8]. Pixilation involves sub-sampling, it reduces the number of distinct pixel values in an image by replacing a square block of pixel values with their averaged. Whereas smoothening of the edges in an image is defined as Blurring [4,5].

Another existing technique is by placing a bar mask over the eyes, which was later modified by extending the mask by covering the nose as well (known as T mask) represented in figure. Though this seemed privacy preserving for the naïve human eye, but the re-identification software's would identify such images with fair percentages as they considered positioning of another facial traits on the scale of pixels, resulting in literally complete match even after applying these Bar and T masks.

Threshold is among another used adhoc methods where colored pixels are changed into only white or black colored pixels. More is the value of threshold greater is the percentage match of correctness [20].

The model 'PrivacyCam' proposed by Senior et al. [6] secures the access to video stream using cryptography and suppresses foreground images using these techniques. Newton et al. [8] introduced a model based on k-anonymity, called k-same. This model guarantees the de-identified face to be similar with at least k faces in given dataset, thereby limiting the face recognition performance by 1/k. An extension of k-same was introduced as k-same-select [7] that ensured the gender privacy, but since these all techniques used appearance based de-identification that is Pixelization, hence are not able to ensure privacy. It is shown by Gross et al. [10] that these techniques based on generating de-identification images by just distorting the pixels are still prone to privacy attacks as some humans could still be identified.

A different approach by reducing the number of Eigen vectors from the basis vectors used in construction of face was proposed by Phillips [9] for ensuring privacy in face detection. A method was introduced by Gross et al. [10], in which they generalize the facial characters using three points such as position of center of eyes, nose and mouth and hence making their resulting images according to that generalized location of points. Later, Gross et al. [11] described a fitting algorithm for joint parameters which improves significantly upon standard alternating fitting algorithm as it considers linear, bilinear, and quadratic models into one formulation.

### *B. Analysing Movement Along a Path*

A surveillance data could also possibly contain several events or people moving in various directions at any given time. Our task is to find the most preferable and suitable path leading to preferred destinations for most of people. We have various crowd counting algorithms that constitute of three paradigms: Pedestrian Detection, Visual Feature Trajectory, and Feature based regression. These algorithms are based on the motion features and focus on whole group and hence tend to suffer in very crowded scenarios [12]. A Solution for this problem is provided by Chan et el. [13]. Their algorithm first segments the crowd into components of homogeneous motion followed by the comparison of number of per segment to the extracted holistic features of that segmented region.

To detect the movement of people inside a building Kim and Davis [14] proposed a multi-view multi-hypothesis approach for segmenting and tracking multiple people on same ground plan. For detecting motion on trajectories using single camera Wu et el. [15] worked on detection, recognition and representation of moving events. Liu and Lai [16] proposed a summarization method for visualizing trajectories of suspicious people in a static image across multiple cameras in a building.

Lee and Kim [20] developed a relatively low cost background subtraction model using background sets with image- and colour-space reduction. The image space here deals with the foreground part having dynamically moving items whereas the background sets are used to detect the information about the surrounding background environment like trees, buildings and roads etc. with respective to the scenario.

Zhang et el. [21] studied the automatic fall detection of elderly people living in old age homes using the depth perception Kinect cameras and simple RGB cameras studying the positions of major body joints and determining whether the person has fallen or not by comparing the difference in movement of the joint positions at various time intervals. There model preserved privacy using the depth perception cameras, building designs and found that the sizes of doors and emergency exits have a straight relation with evacuation efficiency. They found evacuation efficiency in small building but was of very less use when applied using the conventional cameras.

### *C. Studying Evacuation Models*

During emergencies, panic situations can cause injuries and casualties as everyone rushes to save themselves, resulting in stampedes, since every person is trying to use a narrow trajectory (emergency exit stairs) at that interval of time. Ha and Lykotrafitis [16] studied agent based modeling for is more by using small doors. Pelechano and Badler [17] designed a model that animates the escape routes followed by people who are familiar as well as unfamiliar with that respective building in case of emergency. Ma et al. [2] studied the escape routes of ultrahigh rise buildings like Twin Towers after 9/11. They considered various cases involving study of only one person evacuating from top floor taking consideration of general characteristics like mean speed and total time needed for evacuation.

These models for studying escape routes do not consider factor of maintaining privacy of real world people. Our objective is to supply anonymized image data to data analysts for research purposes that require study of a general trend of the crowd, like designing better evacuation route plans to allow for more efficient buildings.

*Figure 1: Represents original input image*

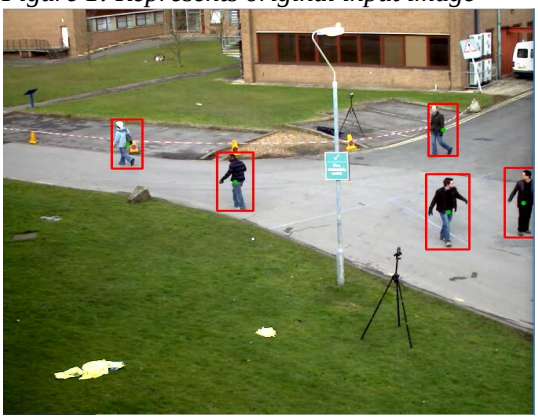


*Figure 3(a): Represents Contours Output*

*Figure 3(b): Represents Blob Output*

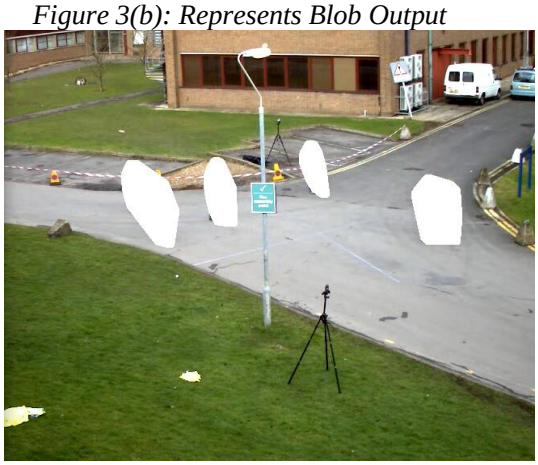


*Figure 2(a) : Represents Contour mask*

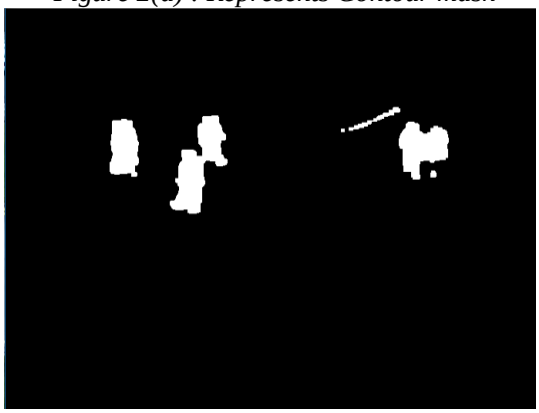
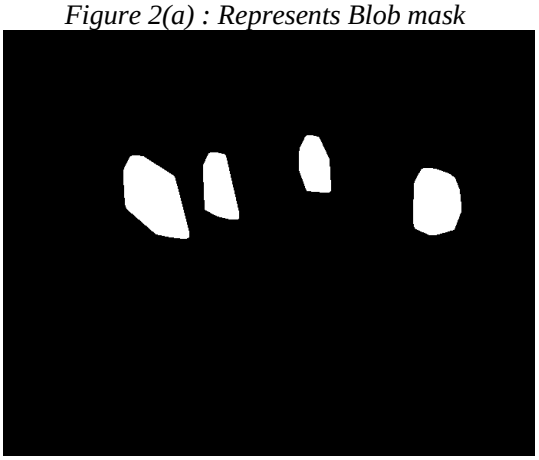


*Figure 2(a) : Represents Blob mask*



## IV. Methodology

Our goal is to preserve the majority of background to achieve high utility rate without compromising the privacy of any individual in the video stream. In this process, the CCTV video feed is analyzed frame by frame, and all changes are made to the individual frames, which are then combined to make an output video stream. The changes to each frame allow us to delete out the private elements in the foreground.

We start by differentiating between the private and public content of the image by identifying the background and foreground as two different entities. This is done by taking two adjacent frames ($n^{th}$ and $(n+1)^{th}$ frames) and analyzing them together to obtain the background and the foreground layers. Each of the frames is then scanned to detect moving objects in the video. The objects in motion are marked by drawing a rectangular box around them and making a center point for reference as shown in Figure 1. In the next step, a mask containing the information of the private data of the same dimension is generated from the foreground layer as shown in Figure 2. Since there is no preprocessing of image involved with the image stream, the overall processing time is very low (in milliseconds) allowing this system to work in real-time. This also allows the system to achieve high level of scalability.

### *1. Generating the mask*

We use two methods to create the mask: Contour and Blob. We explain about each of them in detail after this section.

### *1.1 Using Contours*

Contours are defined as a curve joining all the continuous points (along the boundary), having same

color or intensity. They are a useful tool for shape analysis and object detection and recognition [22]. In this approach, the system looks for contours in the background layer and marks them to create the mask. This mask is used to create mask grid that s further used to maintain privacy as explained above. This method has a reasonable balance of privacy and utility. As we can see from Figure 2(a), we easily distinguish multiple numbers of individuals from the output file. This approach is focused more towards utility and features sufficient level of privacy required in most use cases. However, this approach may not be well suited to cases requiring maximum privacy (such as studying the design of defense buildings). To cater to such cases, we develop a second approach discussed below.

### *1.2 Using Blob*

Informally, a blob is a region of an image in which some properties are constant or approximately constant; all the points in a blob can be considered in some sense to be similar to each other [23]. In this approach, we detect blobs in the input image data and produce a mask that is covers much more of the private data. It clubs together multiple numbers of blobs that are close to each other to ensure total privacy as shown in Figure 2(b). The mask generated from this approach, therefore, has a higher amount of white space (private data masking) than the Contour approach. This results in a larger area of output image data being masked away affecting the utility of the data. This approach gives maximum priority to privacy preservation in data while retaining reasonable utility.

In the final steps, we draw a grid on the mask to get the location of private data in the image. We call this grid the mask-grid. The mask-grid is superimposed on the original frame and all the private data is deleted from the original image based on the information from mask grid. Solid white color is added over the deleted area to allow for further manipulation (if required) and good aesthetics as represented in Figure 3(a) and Figure 3(b).

The output data can still be used for analysis of building design, but it would require use of a prediction based algorithm to achieve same levels of utility as the Contours approach.

## V. EVALUATION

We implemented this system using Visual Studio 2017 and Open CV 3.3.1 on a 64-bit Windows machine running Windows 10.

### *A. Simulation*

We used a publically available CCTV video feed to check the system performance and to ensure the error-free working of the system. A block of 10 frames was taken from the CCTV video to simulate the results. The simulation was run 15 times using Contour and Blob approaches and the accuracy was evaluated by manually tallying the data in the output image to that of the input image. In these 15 iterations, the system accuracy was in the range of 89%-92% as shown in graph 1. Minor tweaks were done to the code to further improve accuracy in the experimental phase.

### *B. Experiment Setup*

The system input was taken from a Lorex CCTV camera mounted high on a wall inside a room. The feed from this camera was input to the system in real time and outputs using Contour and Blob approach were saved and displayed. Accuracy measurements, using manual tally, were done on 10 random 2-minute clips. In each clip, crowd size varied from 2 to 15 individuals in a single frame. The results of this were similar to that of the simulation. This shows that the system is precise in practice.

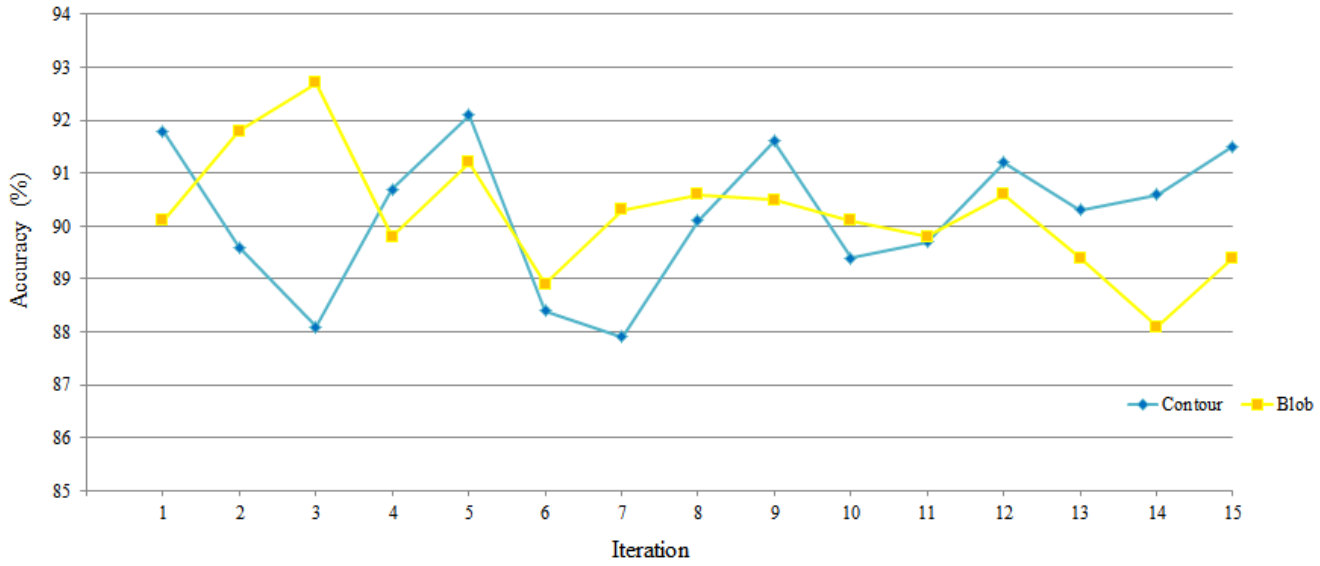
A further experiment was carried out in 10 iterations by varying the frame size for each iteration. The frame size was varied from VGA (640x480) to Full HD (1980x1080). It was observed that there was no correlation between the frame size and the system accuracy as the system displayed consistent levels of accuracy for different variations in frame size. The average accuracy was found to be 89%.

In the last experiment, the frame rate of the video was altered in each iteration. The frame rate was changed from 10fps to 30fps to observe the changes in accuracy. It was observed that the accuracy was lowered as we increase the frame rate as shown in graph 2. The accuracy ranged from 77.6% to 93.8% in Contour and 74.4% to 95.1% for Blob and from this we can infer that the system performed best when the frame rate was around 10fps. This is ideal as the optimal frame rate in most CCTV cameras is 10fps.

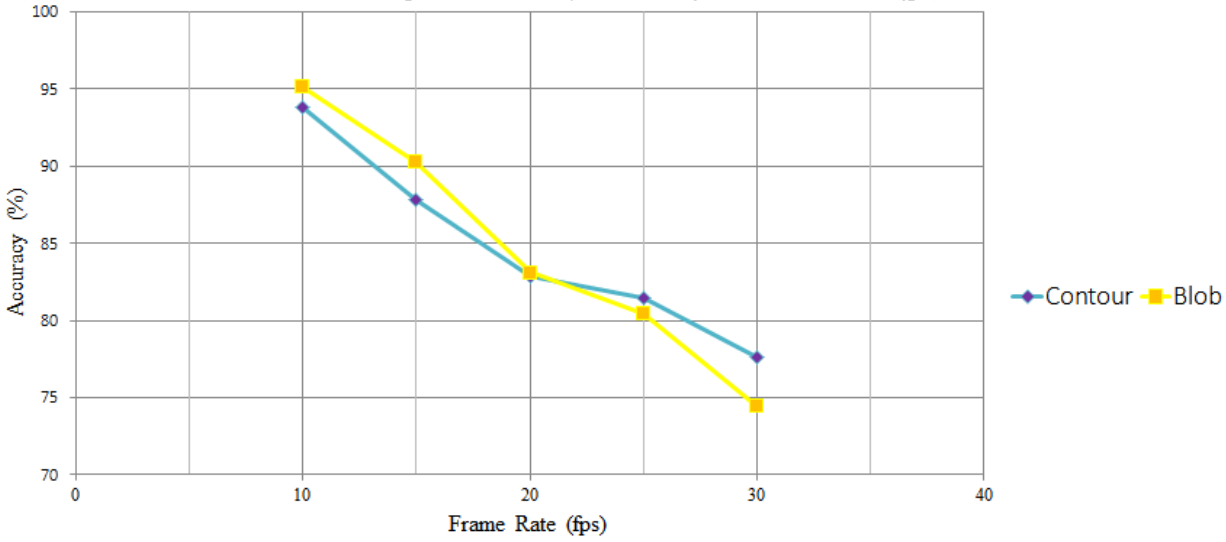
## VI. CONCLUSION

We show that there is lack of system that can ensure high data utility and privacy preservation when analysing image stream data captured by CCTV cameras. This data is most often used while resear-

*Graph 1 : Accuracy vs Iteration rate*




*Graph 2 : Accuracy v/s Change in Frame Rate (fps)*




ching the field of building design. We purpose a system that can efficiently preserve privacy in image data by generating a mask and a mask-grid for each frame. Private data is detected and hidden from the original image data by using this mask and the mask-grid. We derived two approaches to generate the said mask and mask grid: Contours and Blob. Contours focuses on data utility while Blob is more targeted towards data privacy. We demonstrate, through simulation and experiments, that this system is accurate and scalable. At last, we outline the potential applications and extensions to the system in the future work.

## VII. FUTURE WORK

Our proposed model is not very cumbersome. It involves simple variation detection mechanisms to identify the people moving in the CCTV image stream and is able to preserve the identity of them after applying few computational processes. Our model is competent of detecting any moving object, followed by its proper de-identification. The only area where it fails is when the person in the video stream is either sitting or not moving. Since our model judges the foreground with the virtue of movement, hence it fails to privatize the still objects.

This limitation can be eliminated by training our model using the machine learning principles. By means of deep learning a model could be trained for recognizing people or any other object that we need to privatize in a CCTV image stream. By adding this feed from the deep learning model, along with the data from our developed model, we would achieve total fool proof accuracy in identifying and completely preserving the privacy of the required subject in provided image stream.

Similarly if our model is used for monitoring the flow of traffic, it would be able to detect and privatize all the vehicles moving as well as stuck in traffic jams etc., if a training model is used to assist our model in identifying every still or parked vehicles.

Secondly, The IOT sensors such as the Grid Eye Infrared Sensors and the heat sensors etc. can be used along with our model to bring more accuracy in verifying the movement or in purpose of counting the people or objects in the provided CCTV image stream. Lastly, our model can be vividly used in incorporating more of building design technologies, such as path mapping, crowd density and efficient route planning for designing large buildings.

## VIII. REFERENCES


[1] Bensilum, M. and Purser, D.A., 2003. Grid Flow: An Object-oriented Building Evacuation Model Combining Pre-movement And Movement Behaviours For Performance-based Design. Fire Safety Science 7: 941-952. doi:10.3801/IAFSS.FSS.7-941.

[2] J. Ma, W.G. Song, W. Tian, S.M. Lo, G.X. Liao, Experimental study on an ultra high-rise building evacuation in China, In Safety Science, Volume 50, Issue 8, 2012, Pages 1665-1674, ISSN 0925-7535, https://doi.org/10.1016/j.ssci.2011.12.018.

[3] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," in IEEE Pervasive Computing, vol. 2, no. 1, pp. 46-55, Jan-Mar 2003, doi: 10.1109/MPRV.2003.1186725.

[4] M. Boyle, C. Edwards, and S. Greenberg. The effects of filtered video on awareness and privacy. In ACM CSCW, pages 1–10, 2000.

[5] C. Neustaedter, S. Greenberg, and M. Boyle. Blur filtration fails to preserve privacy for home-based video conferencing. ACM TOCHI, 2005.

[6] A. Senior, S. Pankati, A. Hampapur, L. Brown, Y.-L. Tian, and A. Ekin. Blinkering surveillance: enabling video surveillance privacy through computer vision. IEEE Security and Privacy, 3(5), 2005.

[7] R. Gross, L. Sweeney, F. de la Torre and S. Baker, "Model-Based Face De-Identification," 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06), 2006, pp. 161-161. doi: 10.1109/CVPRW.2006.125.

[8] E. Newton, L. Sweeney, and B. Malin. Preserving privacy by de-identifying facial images. IEEE Transactions on Knowledge and Data Engineering, 17(2):232–243, 2005.

[9] P. J. Phillips. Privacy operating charateristic for privacy protection in surveillance applications. In AVBPA, 2005.

[10] R. Gross, E. Airoldi, B. Malin, and L. Sweeney. Integrating utility into face de-identification. In Workshop on Privacy Enhancing Technologies (PET), June 2005.

[11] R. Gross, L. Sweeney, F. de la Torre and S. Baker, "Semi-supervised learning of multi-factor models for face de-identification," 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, 2008, pp. 1-8. doi: 10.1109/CVPR.2008.4587369.

[12] M. Atallah, E. Bertino, A. Elmagarmid, M. Ibrahim, and V. Verykios. Disclosure limitation of sensitive rules. In Proc of IEEE Knowledge and Data Engineering Workshop Workshop (KDEX), pp 45-72, Chicago, Nov 1999.

[13] A. B. Chan, Zhang-Sheng John Liang and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, 2008, pp. 1-7.

[14] K. Kim and L.S. Davis, Multi-camera Tracking and Segmentation of Occluded People on Ground Plane Using Search-Guided Particle Filtering, ECCV, 2006.

[15] G. Wu, Y. Wu, L. Jiao, Y. Wang and E.Y. Chang, Multi-camera Spatio-temporal Fusion and Biased Sequence-data Learning for Security Surveillance, Proceedings of the Eleventh ACM International Conference on Multimedia, 2003.

[16] Vi Ha, George Lykotrafitis, Agent-based modeling of a multi-room multi-floor building emergency evacuation, In Physica A: Statistical Mechanics and its Applications, Volume 391, Issue 8, 2012, Pages 2740-2751, ISSN 0378-4371, https://doi.org/10.1016/j.physa.2011.12.034.

[17] N. Pelechano and N. I. Badler, "Modeling Crowd and Trained Leader Behavior during Building Evacuation," in IEEE Computer Graphics and Applications, vol. 26, no. 6, pp. 80-86, Nov.-Dec. 2006.doi: 10.1109/MCG.2006.133.

[18] GIMP. [Online]. Available: https://www.gimp.org/. [Accessed: 31-Oct-2017].

[19] facedetect: a simple face detector for batch processing. [Online].Available:http://www.thregr.org/~wavexx/s

oftware/facedetect/#blurring-faces-within-an-image. [Accessed:31-Oct-2017].

[20] H. Lee, H. Kim and J. I. Kim, "Background Subtraction Using Background Sets With Image- and Color-Space Reduction," in IEEE Transactions on Multimedia, vol. 18, no. 10, pp. 2093-2103, Oct. 2016. doi: 10.1109/TMM.2016.2595262.

[21] Zhang C., Tian Y., Capezuti E. (2012) Privacy Preserving Automatic Fall Detection for Elderly Using RGBD Cameras. In: Miesenberger K., Karshmer A., Penaz P., Zagler W. (eds) Computers Helping People with Special Needs. ICCHP 2012. Lecture Notes in Computer Science, vol 7382. Springer, Berlin, Heidelberg.

[22] "OpenCV: Contours : Getting Started", Docs.opencv.org, 2017.[Online].Available:http://docs.opencv.org/3.3.1/d4/d73/tutorial_py_contours begin.html. [Accessed: 12- Dec- 2017].

[23] "Blob detection", En.wikipedia.org, 2017. [Online]. Available: http://en.wikipedia.org/wiki/Blob_detection. [Accessed: 12- Dec- 2017].

[24] Microsoft Visual Studio. (2017, December 5). In Wikipedia, The Free Encyclopedia. Retrieved 10:36, December12,2017, from https://en.wikipedia.org/w/index.php?title=Microsoft_Visual_Studio&oldid=813782340.

[25] OpenCV. (2017, November 22). In Wikipedia, The Free Encyclopedia. Retrieved 19:50, December 12, 2017, from https://en.wikipedia.org/w/index.php?title=OpenCV&oldid=811519079.

[26] Pulli, Kari; Baksheev, Anatoly; Kornyakov, Kirill; Eruhimov, Victor (1 April 2012). "Realtime Computer Vision with OpenCV", Queue. pp. 40:40–40:56. doi:10.1145/2181796.2206309.

[27] Itseez leads the development of the renowned computer vision library OpenCV. http://itseez.com