K-Nearest Neighbours Based Classifiers for Moving Object Trajectories Reconstruction

Muhammad Majid Afzal
Decibel Insight
London, UK
email: admtel@gmail.com

Karim Ouazzane, Vassil Vassilev
Cyber Security Research Centre
London Metropolitan University, London, UK
email: {k.ouazzane, v.vassilev}@londonmet.ac.uk

***Abstract*—This article presents an exemplary prototype implementation of an Application Programming Interface (API) for incremental reconstruction of the trajectories of moving objects captured by Closed-Circuit Television (CCTV) and High-Definition Television (HDTV) cameras based on K-Nearest Neighbor (KNN) classifiers. This paper proposes a model-driven approach for trajectory reconstruction based on machine learning algorithms which is more efficient than other approaches for dynamic tracking, such as RGB-D (Red, Green and Red Color model with Depth) images or scale or rotation approaches. The existing approaches typically need a low-level information from the input video stream but the environment factors (indoor light, outdoor light) affect the results. The use of a predefined model allows to avoid this since the data is naturally filtered. Experiments on different input video streams demonstrate that the proposed approach is efficient for solving the tracking of moving objects in input streams in real time because it needs less granular information from the input stream. The research reported here is part of a research program of the Cyber Security Research Centre of London Metropolitan University for real-time video analytics with applicability to surveillance in security, disaster recovery and safety management, and customer insight.**

Keywords-Video surveillance; Real-time video analytics; Model-driven motion description; Moving objects tracking; Trajectory reconstruction; Incremental algorithms; Machine learning.

I. INTRODUCTION

Several different approaches have been used for moving objects tracking but this remains a difficult issue in computer vision and video analytics. Multiple objects tracking have many useful applications in the scene analysis for computerized surveillance. If the system can track different objects in an environment of multiple moving objects and reconstruct their trajectories, then there will be a variety of applications, such as motion detection/tracking in secure areas, controlling the flow of mass movements, analysis the pattern of movements etc. This research is focused on reconstructing the trajectory of body movements in continuous stream of video signals with the help of classifiers for the purpose of further analysis. The existing approaches [1]-[4] typically need a low-level information from the input video stream but the environment factors (indoor light, outdoor light) affect the results. The use of classifiers would make the object tracking simpler and more efficient. In this research, KNN has been selected as an algorithm for classification because it is simple and efficient and fulfills the requirements. Our method is based on the use of a predefined body model to capture only the most relevant information needed to reconstruct the trajectory. This approach has not been explored much by the research community - see [1][2] for use of approximate proximal gradient and Gaussian mixture model for object tracking, [3][4] for the use of detection and tracking approach, [5][6] for data association done with the help of online learning and [7][8] for interoperability of traditional trajectory information and generic sensors.

This research is part of the research program for Simulation-based Visual Analysis of Individual and Group Dynamic Behavior carried out within the Cyber Security Research Centre of London Metropolitan University. The research group is interested in real-time video analytics with applicability to surveillance in security, disaster recovery and safety management, and customer insight. The ultimate goal of this research program is to construct an efficient framework for visual analytics in real time, as presented in [17].

In our approach, moving object tracking is based on the object-centric representation of the position which forms a tube-like model of the spatial navigation and allows isolated manipulation of the video objects within the focus [10]. This can be achieved through an incremental algorithm for processing of the information flow, as illustrated in Figure 1.

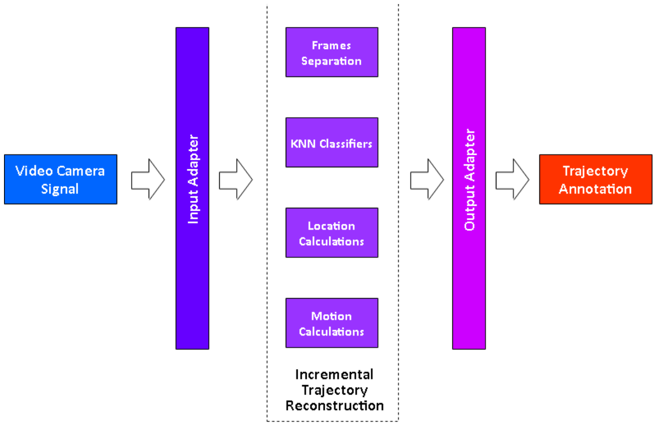

Figure 1. Incremental trajectory reconstruction using KNN classifiers

The moving human object in the video is modeled as a collection of spatiotemporal object volumes (object tubes). Key for reconstructing of the trajectory in this model is the estimation of the object positions and the navigation parameters of the object movements such as rotation, direction of movement and speed.

KNN classifiers are used for reconstruction of moving object trajectories and they help in starting the extraction of the motion information from the video and representation of object trajectories in a 3D grid. Motion based on video representations has been used in other video navigation and annotation systems, but the focus of these systems is mainly on providing an in-scene direct moving object trajectory from the video. As expected, the reconstruction of the trajectory is based on analytical methods for connecting the spatial locations of the identified objects across the frames. This is pursued on the basis of incremental approximation of the spatial locations of the video frames using different computational techniques and approximations.

The rest of this paper is organized as follows. Section II describes the proposed classifiers methodology. Section III addresses the data post processing. Section IV reports on the implementation of the framework. Section V presents the experimental evaluation. The conclusions and references close the article.

II. USING CLASSIFIERS FOR RECOGNITION AND TRACKING

This section shows the use of the classifiers for segmentation of moving objects based on the features extracted from the input video stream. The feature vectors are created at the learning stage, as displayed in Figure 2.

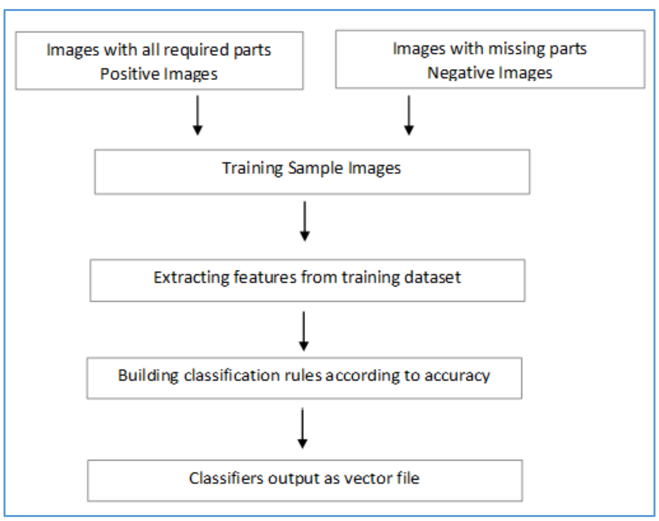

Figure 2. Classifiers learning steps

The process can be explained using a series of equations, calculated at each step. They lead to the formation of the feature matrix used by the classifiers.

Let us assume the input video stream containing all features and data can be described as follows:

$$A = [\, a_1; a_2;\ a_3; \dots a_n] \in \mathfrak{R}^{n*m} \tag{1}$$

The above equation describes the input data as a multi-dimensional matrix with m as the number of features and n as the number of samples. The j[st] sample is

$$a_j \in \mathfrak{R}^{1*m} \tag{2}$$

while the j[st] feature vector is

$$f_j (j = 1, \dots m) \tag{3}$$

In accordance with this, the multi-dimensional matrix of combined features and samples takes the form

$$A = f_1; f_2;\ f_3; \dots f_m \tag{4}$$

For a matrix C, the Frobenius norm can be calculated as

$$||C||_{Fi} = \sqrt{\sum_{i=i}^{n} ||c^i||^2_{\ 2}} \tag{5}$$

$$||C||_{Fj} = \sqrt{\sum_{j=i}^{m} ||c_j||^2_{\ 2}} \tag{6}$$

Using this measure, the features can be shown as

$$||C||_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=i}^{m} c_{ij}^2} \tag{7}$$

where, c^i and c^j denote a row and a column of the original multi-dimensional matrix, respectively. This matrix contains all information for the features used by the classifier. To estimate a single feature f_j, we can use the following linear regression model:

$$f_j \approx \sum_{i=1}^{m} f_i\, s_{i,j} = A_{sj}, \in j = 1,2, \dots, m \tag{8}$$

where, $s_{i,j}$ represents the i[th] feature vector to the j[th] sample. In this case the co-efficient vector of the feature f_j, can be formulated as

$$s_j = \left[\, s_{1,j}; \dots; s_{ij}; \dots\ s_{m,j} \right] \in \mathfrak{R}^{m+1} \tag{9}$$

As a result, the multi-dimensional matrix can be written as

$$A \approx AS \tag{10}$$

where A is the linear combination of all features and

$$S = [s^1; \dots, s^j; \dots; s^m\ ] \in \mathfrak{R}^{m+m} \tag{11}$$

The value of S can be calculated as follows:

$$min\ ||A - AS||_F^2 \tag{12}$$

To reduce the redundancy and keep the features unique, we can use the co-efficient matrix of $| < s^i\ ,\ s^j < |$, where, s^i and s^j denote i[th] row and j[th] row vector of S, respectively. To use all vectors, the following formulas hold:

$$\Omega(S) = \sum_{i=1}^{m} \sum_{j=1, j\neq 1}^{m} | < s^i\ , \quad s^j > | \tag{13}$$

$$\Omega(S) = \sum_{i=1}^{m} \sum_{j=1}^{m} |< s^i\ ,\ s^j >| - \sum_{i=1}^{m} | < s^i\ ,\ s^j > | \tag{14}$$

$$\Omega(S) = \sum_{i=1}^{m} \sum_{j=1}^{m} |< s^i\ , s^j >| - \sum_{i=1}^{m} \left\| s^i \right\|_2^2 \tag{15}$$

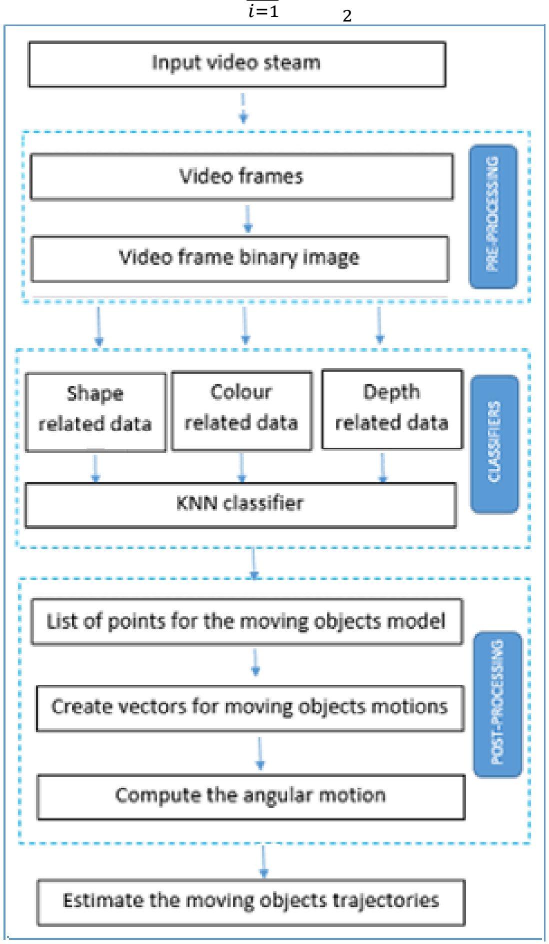

Figure 3. Flow of video stream analyzed using KNN classifiers

The values calculated using (13) are required to identify the features in the input video stream and to track the moving objects and their parts. These features will be used by the classifier at the later stage to reconstruct the trajectories of moving objects. This process is executed in a sequence of steps, as shown in Figure 3.

Features information generated with the help of the equations presented in this section and the KNN classifier decide if a moving object is a human being or not. Similarly, classifiers decide about different moving parts of a moving object.

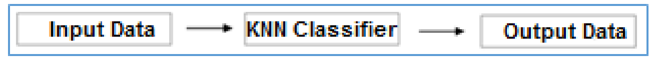

Figure 4. Classifiers execution and extracting information steps

III. DATA POST PROCESSING

In order to provide informative reconstruction of the trajectories, it is essential to perform some post processing of the data generated after the classifier completes its task. The most important processing steps are as follows:

- **Estimating the viewing direction:** The viewing direction is calculated with the help of the head sphere of the moving object model and with the position of the eyes in the head sphere. If the eyes direction and moving object direction is same then object is viewing in direction of movement.
- **Orientation of the moving parts:** This information is calculated with the help of position of face and head hairs. This step is necessary in order to distinguish between left and right hand. The same is applied on the legs of moving object.
- **Completing the invisible body parts:** The missing body parts of moving object of seven sphere based model are estimated in order to generate meaningful trajectory data.
- **Estimating the depth of 2D projection:** The depth of moving object in the video stream is calculated with the help of geometric calculations.
- **Detecting of the moving objects:** The moving objects can be detected with the help of some historical information. All static objects do not change the position in a sequence of frames, while the dynamic object do and this can be a criteria for identifying new objects on the scene.
- **Origin adjustment:** The logical center of the scene can be adjusted in order to make the displacement and movement calculations easier
- **Camera position adjustment:** The camera position can be adjusted to coincide with the origin of the visual scene.

The above tasks are executed after the trajectory data is calculated using the information obtained during the trajectory reconstruction to facilitate the further analysis by the behavior analyzer of the video analytics framework [18]. The limited space of the article does not allow more details.

IV. IMPLEMENTATION OF THE FRAMEWORK

The trajectory reconstruction module of the video analytics framework performs the actual processing of the video frames under the control of **OpenCV** engine [11]. The engine supports the following main operations:

- High-level GUI and Media I/O
- Image processing of the video frames

- Geometric transformations
- Structural analysis and shape approximation

Our module operates in real-time, implementing recurrent algorithm for KNN classification and trajectory reconstruction based on the model described in the paper. It performs several tasks as follows:

A. Selection of video frames for processing

The video data consists of video frames which are 2D objects. These frames are combined in the time sequence to form a video by the digital devices as shown in Figure 4.

Typically, the CCTV and HDTV surveillance cameras produce frames at a rate which does not exceed thirty frames per second. Most of the video processing frameworks also do not process each and every video frame. Some of the frames presented in Figure 5 are shown in white color and few more are shown in gray color as we assume that we are processing only the frames in grey after skipping few frames in white. The criteria for choosing which frames to process depends on the complexity of the algorithms and the frame content.

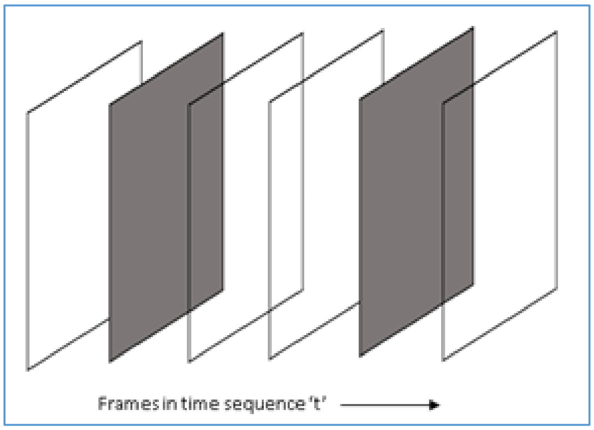

Figure 5. Sequence of frames

B. Moving objects segmentation using classifiers

This component of the trajectory reconstruction module performs operations on all selected frames to identify and approximate the contour of the objects within the frame (Figure 6). The Input video stream data is provided to the classifiers to distinguish the moving object in focus. The segmentation component first converts the frame into binary format and then performs processing of the pixels to find the approximate contour of the moving object.

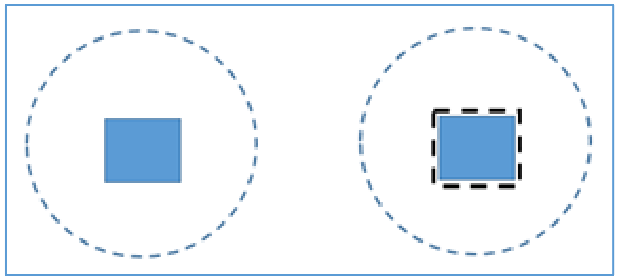

Figure 6. Shaping its projections on the frame using classifiers

C. Computing moving objects displacement

Displacement component keeps track of the moving object identified by the segmentation component of the module. It calculates the displacement of the moving objects in each processed frame, which is needed for subsequent trajectory reconstruction.

D. Reconstructing the moving objects trajectory

The reconstructed trajectory data is calculated on the basis of the information about object location, their descriptors and the values of displacement. It is a continuous stream of information calculated recurrently and generated as an output of the module for further analysis.

V. EXPERIMENTAL EVALUATION

In this section, we carry out simulated experiments to demonstrate the advantage of the proposed KNN based classifier for reconstruction of trajectories compared with other three approaches namely CEMMT [15], DCOMT [16] and KSP[17] To evaluate the performance of different approaches, two most commonly used datasets PETS 2009 S2L1 and PETS 2009 S3MF1 are selected. These datasets have different challenges, such as occlusion, people with same color of clothing, pose changes and exit and entry of scene.

To compare the multi object tracking algorithms, we have adopted the CLEAR metrics [14] which is the most widely used protocol for quantitative evaluation. The different measures for comparison in this benchmark are as follows:

TABLE I. COMPARISON VALUES OF PETS 2009 S2L1

Methods	**Comparison Values**							
	Rec.	***Prec.***	***GT***	***MT***	***ML***	***IDs***	***MOTA***	***MOTP***
CEMMT [15]	94.2	98.4	23	21	1	11	90.6	80.2
DCOMT [16]	90.0	98.7	23	19	0	18	88.3	79.6
KNN	**85.9**	**97.6**	**23**	**6**	**0**	**2**	**82.6**	**90.1**

TABLE II. COMPARISON VALUES OF PETS 2009 S3MF1

Methods	**Comparison Values**							
	Rec.	***Prec.***	***GT***	***MT***	***ML***	***IDs***	***MOTA***	***MOTP***

Methods	Comparison Values							
	Rec.	***Prec.***	***GT***	***MT***	***ML***	***IDs***	***MOTA***	***MOTP***
CEMMT [15]	97.7	99.4	7	7	0	0	97.1	83.4
KSP[17]	87.9	95.4	7	6	1	0	83.7	77.8
KNN	**96.8**	**98.7**	**7**	**7**	**0**	**0**	**95.6**	**94.7**

Groundtruth (GT): The number of trajectories in the groundtruth.

Mostly tracked trajectories (MT): The percentage of trajectories that are successfully tracked for more than 80 percent divided by ground truth.

Mostly lost trajectories (ML): The ratio of mostly lost trajectories, which are successfully tracked for less than 20 percent.

Partially tracked trajectories (PT): The ratio of partially tracked trajectories.

ID switches (IDS): The total number of times that a tracked trajectory changes its matched groundtruth identity.

Recall (Rec.): The number of correctly matched detections divided by the total number of detections in groundtruth.

Precision (Prec.): The number of correctly matched detections divided by the number of output detections.

Multi-Object Tracking Accuracy (MOTA): A measure of tracking accuracy that takes into consideration, false positive, false negatives and ID switches

Multi-Object Tracking Precision (MOTP): This measures the position of objects in experimental results with the actual dataset.

A. *Quantitative evaluation*

Table I shows the experiment comparison values of PETS 2009 S2L1 dataset. This is a difficult dataset as it has 794 frames. Moving objects (people) in the dataset are wearing same color cloths. Dataset has three different backgrounds house, grass and street. As shown in Table I, this dataset is used with different object tracking algorithms,

- CEMMT [15] generate multiple few hypothesis for each detection and selecting those which have minimize energy, in this way moving object tracking is the minimization of continuous energy
- DCOMT [16] simple closed form solution is used as continuous fitting problem for trajectory estimation

Our approach outperforms the other methods in terms of ID switches and MOTP. CEMMT [15] obtained the best results in terms of Recall (94.2), MT (21) and MOTA (90.6) but has more ID switches than our method [11]. Best precision (98.7) value is obtained by DCOMT [16]. Figure 6 shows the comparison of the values obtained by using different methods during the experiments. It is clear from the graph that DCOMT [16] has high number of ID switching while our approach has low ID switching. Our approach also outperforms MOTP.

Table II shows the experiment comparison values of PETS 2009 S3MF1 dataset. This dataset has 107 frames. Dataset has three different backgrounds house, grass and street like the previous dataset. Initially, objects are moving in uniform direction in this dataset and then objects start motion in random directions. As shown in Table II, this dataset is used with different object tracking algorithms in the same way as previous table. Our approach obtains the best results in terms of multi object tracking accuracy with the difference of 11.4 percent.

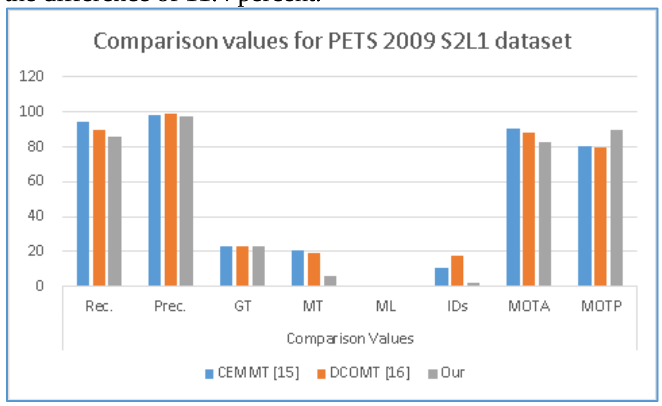

Figure 7. Comparison of the values for PETS 2009 S2L1 dataset (Y-Axis is showing the percentage)

Relatively to CEMMT [15] it is a bit better in recall (0.9 percent) and precision (0.7 percent). This is visible from the graph in Figure 8. Figure 8 shows clearly that MOTP of our approach is better than the other two approaches.

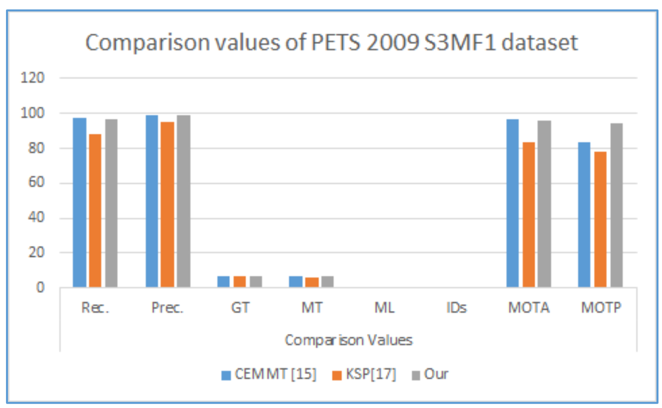

Figure 8. Comparison of the values for PETS 2009 S3MF1 dataset (Y-Axis is showing the percentage)

B. *Qualitative evaluation*

We applied our framework to PETS 2009 S2L1 dataset. Figure 9 shows the changing frames with tracking of several moving objects identified on them. In Figure 8, the trajectory of objects with ID=9 and ID=1 occupy two different positions in frame 290. After five frames in frame 295 object with ID=9 covers object with ID=1.

Figure 9. PETS 2009 S2L1 dataset (frame number 290, 295 and 319)

However, object with ID=1 does not lose its trajectory and there is no ID switch. Finally, in frame 319, object with ID=1 does not cover object with ID=9 anymore, its direction of movement has changed and the trajectories split. This shows fewer ID switches even the moving objects were overlapping.

Dataset PETS 2009 S3MF1 is used with our approach and Figure 10 below shows the tracking of new moving objects entering the scene.

Figure 10. PETS 2009 S3MF1 dataset (frame number 38 and 68)

Figure 10 shows that ID=6 and ID=7 are entering in view in frame number 38. In frame number 68, ID=6 and ID=7 have complete tracking information and they show two different trajectories. This shows that our method is also able to track the motion and handle the trajectories of new objects entering the scene.

VI. CONCLUSION

This paper presents an efficient model-driven approach to moving object trajectory reconstruction using KNN classifiers which can be used for real-time video analytics. Our approach has a number of advantages compared to other existing approaches including Microsoft Kinect model [12] [13] commonly endorsed in computer games industry. Firstly, the use of classifiers makes the extraction of trajectory data easier and make it possible in real live video stream. Secondly, trajectory data can be reconstructed using less information because of the simpler geometry which lowers the requirements for preliminary visual image processing. Thirdly, the reconstruction of the trajectory is more efficient because of the simpler approximation, which makes this approach preferable for real-time systems. Finally, the overall algorithms of moving object trajectory reconstruction are far simpler than the other algorithms reviewed in the literature and as a result the software which implements them becomes more compact, which allows an easy embedding in other software for visual analytics.

Our immediate plans after finalizing the basic trajectory data extraction is to implement the full trajectory reconstruction module of the video analytics framework, which is needed for further analysis of the dynamic behavior in areas such as customer insight, security and safety management. Furthermore, we are planning to enhance our model through combining features of the seven spheres model used here with the six lines model of Kinect in order to be able to analyze gestures as well.

REFERENCES

[1] N. Aslam and V. Sharma, "Foreground Detection of Moving Object Using Gaussian Mixture Model", In Int. Conf. on Communication and Signal Processing, IEEE, PP. 1071-1074, Chennai, India, 6-8 April 2017.

[2] H. Masood et al., "A novel technique for recognition and tracking of moving objects based on E-MACH and proximate gradient (PG) filters", In Int. Conf. of Computer and Information Technology (ICCIT), IEEE, pp. 828-839, 22-24 December, 2017, Dhaka, Bangladesh.

[3] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking", In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'12), Providence, RI, USA, pp. 1926-1933, June, 2012.

[4] W. Du and J. Piater, "Tracking by cluster analysis of feature points and multiple particle filters", In Int. Conf. on Pattern Recognition and Image Analysis (ICIAR'05), Toronto, Canada, pp. 701-710, 28-30 September, 2005.

[5] B. Yang and R. Nevatia, "An online learned CRF model for multi-target tracking", In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12), Providence, RI, USA, pp. 2034-2041, 16-21 June, 2012.

[6] G. Shu, A. Dehghan, and M. Shah, "Part-based multiple-person tracking with partial occlusion handling", In Proc.IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'12), Providence, RI, USA, pp. 1815-1821, 16-21 June, 2012.

[7] A. M. Okamura, C. Simone, and M. D. O'Leary, "Force modeling for needle insertion into soft tissue," IEEE Trans. Biomed. Eng., vol. 51, no. 10, pp. 1707-1716, Oct. 2004.

[8] L. Zhang, X. Wen, W. Zheng, and B. Wang , "An Algorithm for Moving Semantic Objects Trajectories Detection in Video", In Proc. IEEE Conf. on Information Theory and Information Security (ICITIS), pp. 34-27, 2010.

[9] A. Boulmakoul, L. Karim, A. Elbouziri, and A. Lbath, "A System Architecture for Heterogeneous Moving-Object Trajectory Metamodel Using Generic Sensors: Tracking Airport Security Case Study", IEEE Systems Journal, vol. 9, pp. 28-291, March 2015.

[10] M. Afzal, K. Ouazzane, V. Vassilev, and Y. Patel, "Incremental Reconstruction of Moving Object Trajectory", in Int. Conf. on Applications and Systems of Visual Paradigms (Visual 2016), IARIA, pp. 24-29, 2016.

[11] M. Kim, N, Ling, and L. Song, "Fast single depth intra mode decision for depth map coding in 3D-HEVC", In IEEE Int. Conf. on Multimedia & Expo Workshops (ICMEW), pp. 1-6, June 2015.

[12] OpenCV https://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.4.13/opencv-2.4.13.exe/ [accessed: 20-02-18].

[13] Microsoft, Developing with Kinect, https://developer.microsoft.com/en-us/windows/kinect/ [accessed: 20-02-18].

[14] R. Stiefelhagen, K. Bernardin, R. Bowers, J.S. Garofolo, D. Mostefa, and P. Soundararajan, "The CLEAR 2006 Evaluation," In Proc. Int. Conf. Classification of Events, Activities and Relationships, 2006.

[15] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking", IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 58-72 2014.

[16] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking", In Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'12), Providence, RI, USA, pp. 1926-1933, June 2012.

[17] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua, "Multiple object tracking using k-shortest paths optimization", IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1806-1819, 2011.

[18] P. Gasiorowski, V. Vassilev and K. Ouazzane, “Simulation-based Visual Analysis of Individual and Group Dynamic Behavior”, In Proc. Int. Conf. Image Processing, Computer Vision & Pattern Recognition (IPCV'16), CSREA Press, pp. 303-309, 2016.