Incremental Reconstruction of Moving Object Trajectory

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

Prof. Karim Ouazzane, Dr. Vassil Vassilev, Yogesh Patel
Cyber Security Research Group, School of Computing and Digital Media, London Metropolitan University, London, UK
email: {k.ouazzane, v.vassilev, y.patel}@londonmet.ac.uk

***Abstract*—This article presents a model and a prototype implementation of a technical application programming interface (API) for incremental reconstruction of the moving objects trajectories captured by closed-circuit television (CCTV) and High-definition television (HDTV) cameras. This paper proposes a unique, much simpler model-driven approach which is more efficient than other approaches for dynamic tracking such as Microsoft Kinect. The research reported here is part of a research program of the Cybersecurity Research Group 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; Moving objects tracking; Trajectory reconstruction; Model-driven motion description; Incremental algorithms.

I. INTRODUCTION

Intelligent and prompt moving object tracking is a difficult issue in the computer vision research area. Multiple objects tracking has many useful applications in 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. This research is focused on reconstructing the trajectory of body movements in continuous stream of signals of a video for the purpose of further analysis and extracting information. Our method is based on the use of a moving object ontology to capture a more detailed information about the trajectory. The approach used in this article has not been explored much by the research community – see [1][2] for the use of structure motion description and initial estimations, [3][4] for the preset trajectory information in robotic control and [5][6] 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 of the Cybersecurity Research Group 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 shown in Fig. 1.

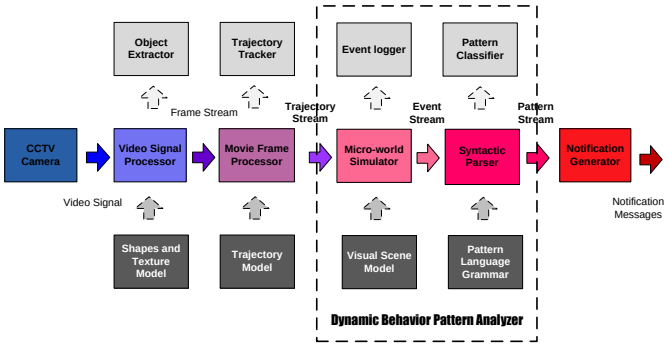

Figure 1. General workflow of the overall framework [8]

Moving object tracking in our approach 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. This can be achieved through an incremental algorithm for processing the flow of information as illustrated in the Fig. 2.

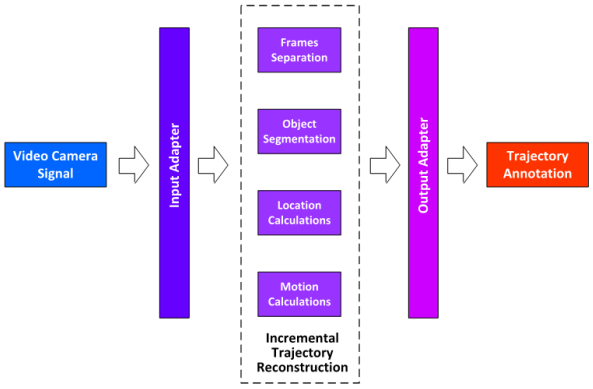

Figure 2. Incremental trajectory reconstruction

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.

Reconstruction of moving object trajectories starts with extraction of the motion information from the video and representation of object trajectories in a 3D grid. Motion based on video representations have 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 foundation of incremental reconstruction of trajectories. Section III describes the method of distance calculation. Section IV addresses the direction estimation. Section V goes into finer details about types of movements. Section VI explains about camera position. Section VII reports about the implementation of framework. The conclusions and references close the article.

II. FOUNDATION FOR INCREMENTAL RECONSTRUCTION OF THE TRAJECTORIES

There are many ways to model the moving objects and they all have different values for the task of trajectory reconstruction. Researchers constantly search for better models in order to make the trajectories' reconstruction process more adequate as well as efficient, so that it can be used in real-time.

A. Comparinsion of the basic geometric shapes

Initially, our research started with a point-based model. After exploring the benefits and limitations of this model, we moved to a more adequate but more complex triangle shape-based model. Fig. 3 shows the trajectory reconstruction with the help of point based model:

Figure 3. Trajectory reconstructed using point based model

The moving object was tracked in the video and annotated graphically by dot shaped labels. Reconstructing of the trajectories with the point-based model is easy and needs less computational resources. The mathematical representation of the points requires only co-ordinates for the dots. It is relatively easy to extract this information from the video in order to reconstruct the trajectory of the moving object.

Point-based model is not appropriate for objects of different size. Also, it is not capable of representing the changes in the shape of object on a move. For instance, the bending and twisting of moving object cannot be represented in point-based model. Therefore, the authors decided to move to a prismatic shape model as it wraps up the whole volume of the physical object body. This is shown in Fig. 4.

Figure 4. Trajectory reconstructed using prismatic model

The prismatic model helps in estimating the size of the object. At the same time, there is a possibility that different prismatic shapes can represent the different parts of the moving object. The prismatic shapes can be combined to create bigger and even different shapes. For instance, a number of prismatic shapes when combined can form a sphere. This approach is widely adopted in computer games where the moving objects are wrapped up in an invisible "capsule".

This model is a step towards estimating the correct size of the moving object. But still it is not possible to cover the motion of the body parts of moving object, for instance, bending, and we are also unable to estimate the twisting of the moving object. To overcome these weaknesses, we obviously need to extend the moving object model through combining multiple shapes, but before that we need to consider its simpler counterpart, the spherical shape model, since it deals with rotations more naturally. Fig. 5 illustrates the trajectory based on spherical model.

Figure 5. Trajectory reconstructed using spherical model

B. Choice of the composite model of the dynamic objects

The starting point of the 3D reconstruction of the object movements using their 2D projections on the frames of a digital video signal is the choice of a suitable composite model. Different approximations are possible depending on the precision needed and the complexity of the recognition algorithms we can afford. For the purpose of the analysis at this stage of the research, we are using *seven spheres model*

(see Figure 6). It is an optimal in the sense that it combines the simplicity of the spherical shape with the sophistication of the composite capsule.

The main parts of the body in this model are represented by separate spherical shapes with one extra sphere to cover the whole body. These six spheres allow tracking of the major type of motions – directed (like *forward*), rotational (like *turning left*), as well as relative to the body (like *bending* and *standing up*). This model can be presented schematically as shown in Fig. 6.

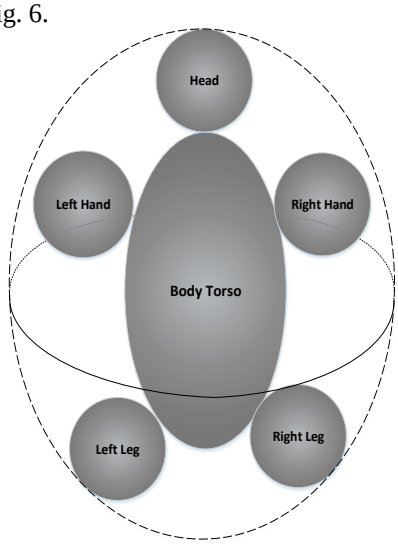

Figure 6. Seven spheres model

Other models are also possible but they all have different benefits and drawbacks from our point of view. The following table summarizes the comparison of some of the popular moving object models.

TABLE I. COMPARISION OF MOVING OBJECT MODELS

Moving object models	Model characteristics			
	Algorithmic Complexity	***Accuracy***	***Representing rotation***	***Amount of information***
Point-based model	Low	Low	No	Low
Spherical model	Low	Medium	No	Medium
Prismatic shape model	Medium	Medium	Yes	Medium
Seven spheres model	Medium	High	Yes	High
Six lines model	High	Highest	Yes	Highest

In future, we do not exclude the possibility to switch to more commonly accepted models such as the line-based similar to the body armature model endorsed by Microsoft in their game platform Kinect (see Fig. 7), but we believe that the seven spheres model is fully adequate for our purpose and has

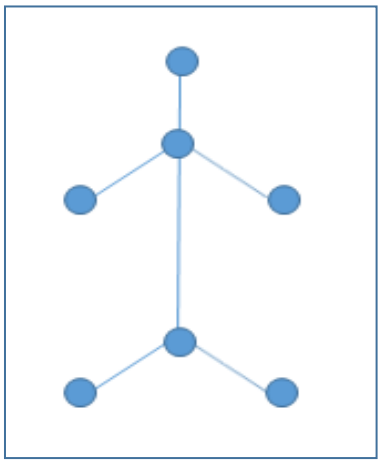

Figure 7. Representation of moving object using dots and lines

additional benefits from computational point of view, since the algorithms for extrapolating the volumes are more efficient.

III. CALCULATING DISTANCES

One of the challenges in this research is to find various distances within the 3D space based on the 2D projections on the video frames - between the camera and the moving object, between the front and the rear of an object etc. This is also known as *depth calculation*. There are some methods already available for this task, but they have their own limitations [8][9]. For instance, they need the camera focal length and other values for preliminary calibration, some methods require constant use of two cameras, etc.

To find a way to overcome this challenge, we have adopted a simple geometrical approach, which calculates the absolute 3D sizes based on their relative 2D projections; as all models used in this research have coordinates values about the moving object then these can be used to find the depth distance. Consider the configuration in Fig. 8:

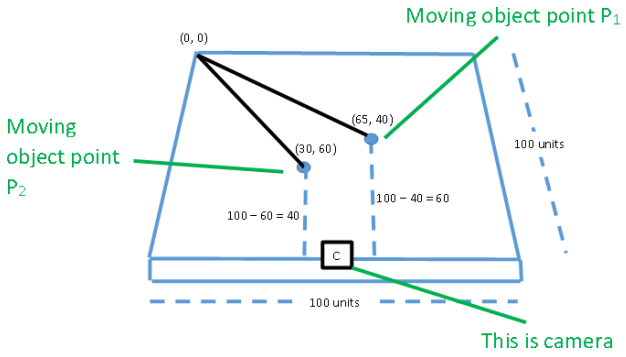

Figure 8. Moving object at point P1 and P2 at two different time instances with depth calculation

In this simple approach and in order to estimate absolute sizes, all what we need are the coordinates of the projections, assuming the position of the camera is fixed in the middle of

the observable space. The advantage of this approach is in its simplicity and independence on any preparatory work although it may lead to more complex algorithms in the case of multiple cameras observing the same space.

IV. ESTIMATING DIRECTIONS

Two main directions need to be estimated for the purpose of the behavior analysis in video analytics – *navigation* (moving direction) and *line of sight* (viewing direction). The direction of movement is estimated entirely using analytical methods which use a simple geometry of space. As far as the viewing direction, we use a combinsation of statistical and analytical methods.

As a first approximation we assume that the viewing direction is the same as the moving direction, at a later stage the viewing direction will be corrected on the basis of the horizontal rotation of the sphere which represents the head. The following possible cases are forming the body of the algorithm for estimating the viewing direction:

- If the body is moving forward along a line then we will estimate the difference between the moving directions of the object over the time interval and if it is stable then the two directions are kept identical. If there is a change caused by rotation of the head, the viewing direction is estimated as the difference between the angle of side rotation and the moving direction.
- If the object is not moving, then the viewing direction is considered to be the difference between the angle of facing the camera and the angle of rotation of the head.
- If the head is rotating in both directions (i.e., oscillating horizontally) the amplitude of the oscillation is estimated to calculate the viewing direction as the middle point.

To implement the above algorithm two parameters have been introduced; the time instance head is turned into a different direction from the moving direction, and the amplitude of oscillation of the head around the viewing direction.

V. TYPE OF MOVEMENTS

In order to provide informative reconstruction of the trajectories, it is essential to track various possible spatial movements, the most important cases are as follows:

Continuous linear movement: The capsula is moving in a direction without rotation. This type of movement generates a simple trajectory.

Static object starts moving: In initial time instance, this static object does not move, however after a few ticks it changes the position. In initial time instance, there is no trajectory for this type of motion but after few time instances the trajectory appears.

Static object moves under the influence of another moving object: This type of motion is similar to the previous but in this case the object motion is caused by another moving object (*knock-down effect*); for instance, a walking person reaches a wall, picks a briefcase from the ground, drops it, etc. In this case the trajectory starts appearing after few time instances.

Moving object disappearing from the scene: At the beginning, the object has a trajectory, but at some point, this trajectory interrupts due to the objects navigating outside of the visual scope of the camera.

Object rotation: This rotation is either the whole moving object rotation or a rotation of some of its parts. This calculation is simpler with the spherical model.

VI. CHANGING THE CAMERA POSITION

All the explanations, figures, and descriptions included in this article are based on the use of two dimensional space as a projection of the three dimensional physical space.

A. Standard Origin

The origin of the two dimensional space is initially considered at the top left (or in three dimensional space, at the top left bottom) corner of the visual scope. The initial position of origin is illustrated in the Fig. 9 where spheres are used to represent the positions.

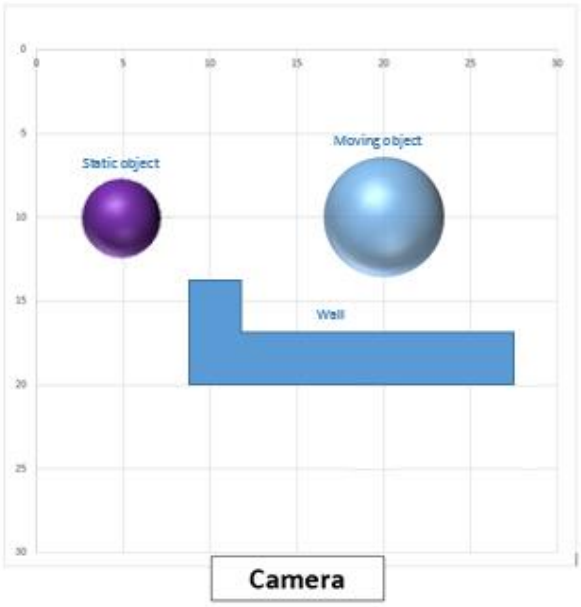

Figure 9. Normal space with origin at the top left corner

We assume that a camera is at the bottom center of the space. The calibration depends upon the space rather than upon the position of the camera so our choice is for convenience only. From the above figure the equation of the origin can be described as follows;

$$O = (x_{O0}, y_{O0}) \quad (1)$$

The position of a camera is represented below,

$$C = (x_{C0},\ y_{C0}) \quad (2)$$

While the position of the sphere is expressed as;

$$S = (x_{S0}, y_{SO}) \quad (3)$$

The vector from the camera to the sphere can be described as;

$$\overrightarrow{CS} = \vec{C} - \vec{S} \quad (4)$$

$$\overrightarrow{CS} = (x_{SO} - x_{c0}, y_{s0} - y_{c0}) \quad (5)$$

The length of the vector can be given as;

$$|\overrightarrow{CS}| = \sqrt{(x_{SO} - x_{c0})^2 + (y_{s0} - y_{c0})^2} \quad (6)$$

B. Shifting the origin at the camera location

To make the whole calculation and explanation simpler, we can move the origin of the space to the position of the camera. This is shown in Fig. 10 where red rectangles denote the updated calibrations.

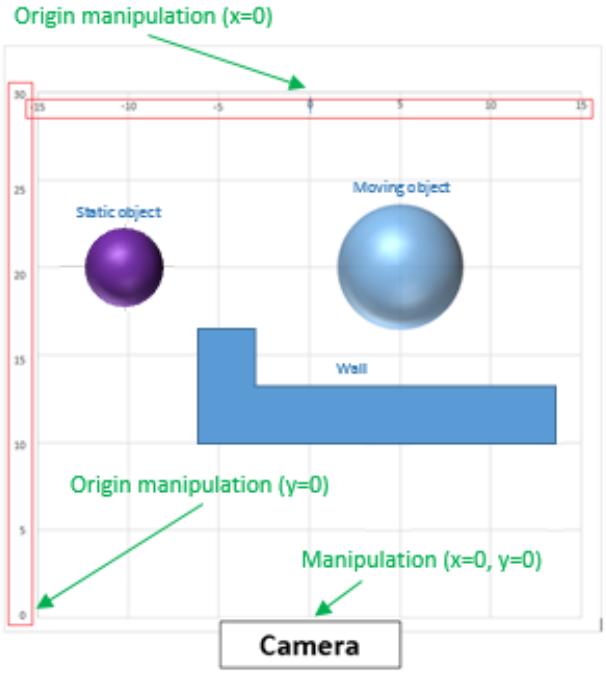

Figure 10. Modified space with the origin at the camera location

The new origin position can be rewritten as below:

$$O' = (x_{o0} + a, y_{o0} + b) \quad (7)$$

where *a* is the displacement value of the origin along X-Axis, while *b* is the displacement value of the origin along Y-Axis.

Due to the above origin displacement, now the camera position is the same as the origin;

$$C' = (x_{o0} + a, y_{o0} + b) \quad (8)$$

The position of the sphere with respect to the new origin is;

$$S' = (x_{s0} - a, y_{s0} - b) \quad (9)$$

By default, the values *a* and *b* should be subtracted from the original value of the sphere. However, if origin displacement involves crossing of the sphere position then the final value should be multiplied by -1. In this case the new vector between a camera and a sphere can be given as below;

$$\overrightarrow{C'S'} = \overrightarrow{C'} - \overrightarrow{S'} \quad (10)$$

$$\overrightarrow{C'S'} = ((x_{s0} - a) - (x_{o0} + a), (y_{s0} - b) - (y_{o0} + b)) \quad (11)$$

Then, the distance between the camera and the sphere is;

$$\overrightarrow{C'S'} = \sqrt{((x_{s0} - a) - (x_{o0} + a))^2 + ((y_{s0} - b) - (y_{o0} + b))^2} \quad (12)$$

All above calculations are relatively simple and can be implemented efficiently which allows the algorithms to be executed in real-time without much difficulties.

VII. Implementation of the Framework

The trajectory reconstruction module of the video analytics framework does the actual processing of the video frames using **OpenCV** open source engine [10]. The module 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

The module operates in real-time using recurrent algorithms based on the model described in the paper. It includes several components which are introduced below;

A. Selection of video frames for processing

Every video data consist of video frames which are 2D graphical objects. These frames are combined in the time sequence to form a video by the digital devices as shown in Fig. 11.

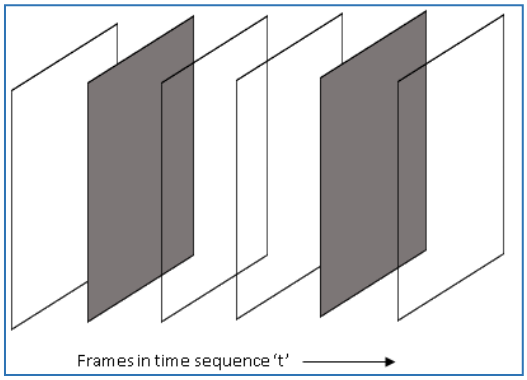

Figure 11. Sequence of frames

Typically, most of the CCTV and HDTV surveillance cameras produce frames at a rate which does not exceed thirty frames per second. The above figure illustrates the flow of frames in a video movie.

Most of the video processing frameworks do not process each and every frame of video data. Some of the frames presented in the above picture are shown in white color and few more are shown in gray color because 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.

B. Moving objects segmentation

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 (Fig. 12).

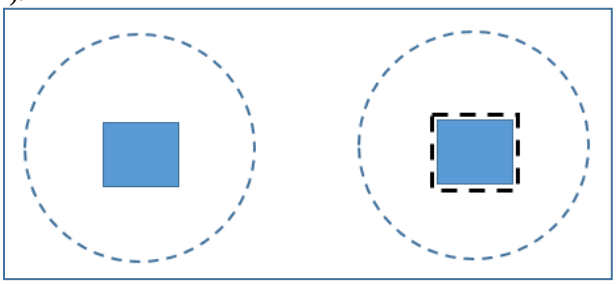

Figure 12. Moving object and sequences of points around the moving object shaping its projection on the frame

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.

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.

VIII. CONCLUSION

This paper presents an efficient model-driven approach to moving object trajectory reconstruction which can be used for real-time video analytics. Our approach has a number of advantages compared to Microsoft Kinect model commonly endorsed in computer games industry [11][12]. First, the trajectory data can be reconstructed using less information because of the simpler geometry which lowers the requirements for preliminary visual image processing. Secondly, the reconstruction of the trajectory is more efficient because of the simpler approximation, which makes this approach preferable for real-time systems. Thirdly, the overall algorithms of moving object trajectory reconstruction are far simpler than the other algorithms reviewed in the literature and the software becomes more compact, which allows an easy embedding in other software for visual analytics.

Our immediate plans, after finalizing the trajectory reconstruction module, is to implement an extension for estimating the viewing direction, which is needed for further analysis of the dynamic behavior patterns in areas such as customer insight. Next, 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] D. Walther, U. Rutishauser, and C. Koch, “On the usefulness of attention for object recognition”, second international workshop on attention and performance in computational vision (WAPCV ’04), pp. 96–103, Conference: Prague, Czech Republic, 2004.

[2] S. Alpert, M. Galun, and A. Brandt, Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration, IEEE transactions on pattern analysis and machine intelligence, pp. 315 – 319, 2012

[3] O. Gerovich, P. Marayong, and A. M. Okamura, "The effect of visual and haptic feedback on computer-assisted needle insertion," Journal of Computer Aided Surgery, pp. 243-249, 2004

[4] R. Alterovitz and K. Goldberg, "Motion Planning in Medicine: Optimization and Simulation Algorithms for Image-Guided Procedures," Springer-Verlag, Berlin Heidelberg, 2008.

[5] 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.

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

[7] 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

[8] P. Gasiorowski, V. Vassilev and K. Ouazzane, “Simulation-based Visual Analysis of Individual and Group Dynamic Behavior”. In: Proc. 20th International Conference on Image Processing, Computer Vision, & Pattern Recognition (IPCV'16), pp. 303-309, 25-28 July, 2016, Las Vegas, USA.

[9] M. Salmistraro, M. Zamarin, L. Raket and S. Forchhammer, “Distributed multi-hypothesis coding of depth maps using texture motion information and optical flow”, IEEE International Conference on Acoustics, Speech and Signal Processing, conference: Vancouver, BC, pp. 1685 - 1689, May 2013.

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

[11] OpenCV library explanation and source https://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.4.13/opencv-2.4.13.exe/download [Last accessed: 20-06-2016]

[12] Developing with Kinect for Windows https://developer.microsoft.com/en-us/windows/kinect/develop [Last accessed: 20-06-2016]