3D Simulation-based Analysis of Individual and Group Dynamic Behaviour in Video Surveillance

Pawel Gasiorowski
The Vinyl Factory Ltd.
London, UK
pavel@thevinylfactory.com

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

***Abstract*—The visual behaviour analysis of individual and group dynamics is a subject of extensive research in both academia and industry. However, despite the recent technological advancements, the problem remains difficult. Most of the approaches concentrate on direct extraction and classification of graphical features from the video feed, analysing the behaviour directly from the source. The major obstacle, which impacts the real-time performance, is the necessity of combining processing of enormous volume of video data with complex symbolic data analysis. In this paper, we present the results of the experimental validation of a new method for dynamic behaviour analysis in visual analytics framework, which has as a core an agent-based, event-driven simulator. Our method utilizes only limited data extracted from the live video to analyse the activities monitored by surveillance cameras. Through combining the ontology of the visual scene, which accounts for the logical features of the observed world, with the patterns of dynamic behaviour, approximating the visual dynamics of the world, the framework allows recognizing the behaviour patterns on the basis of logical events rather than on physical appearance. This approach has several advantages. Firstly, the simulation reduces the complexity of data processing by eliminating the need of precise graphic data. Secondly, the granularity and precision of the analysed behaviour patterns can be controlled by parameters of the simulation itself. The experiments prove in a convincing manner that the simulation generates rich enough data to analyse the dynamic behaviour in real time with sufficient precision, completely adequate for many applications of video surveillance.**

Keywords-Video Surveillance; Video Analytics; Individual and Group Dynamics; Behaviour Patterns; 3D simulation.

I. INTRODUCTION

The analysis of dynamic behaviour has wide applicability in a range of domains, including video surveillance and security, accident and safety management, business customer insight and computer games programming. Of particular interest is the analysis of dynamic behaviour of individuals and groups of individuals moving at relatively normal speeds in bound spaces such as supermarkets, shopping malls, tall buildings, transport stations and airports, large planes and ship vessels.

The recent advancement in visual data processing using numerical methods (Markov models, statistical pattern recognition and qualitative physics for the analysis of individual dynamics [1]-[4] and group dynamics [5]-[7]) as well as the availability of tools for video analysis (e.g. 3VR Video Intelligence Platform, savVI Real-Time Event Detection, PureTechSystems Video Analytics, IndigoVision Advanced Analytics, IBM Intelligent Video Analytics [8]-[12]) show promising results, but the problem still remains difficult.

There are two factors that impact the real-time video analytics: the processing of immense amount of visual data coming from surveillance cameras and the need to associate additional symbolic data with it, in order to conduct the behaviour analysis. While the first issue can be addressed using technological solutions available on the market of tools for visual information processing, the second one remains a serious bottleneck for any video analytics project. Our research forms a central part of the framework currently under development at the Cyber Security Research Centre of London Metropolitan University, dedicated to machine processing of video surveillance information in real time [16]. This framework includes visual scene extraction, trajectory reconstruction, dynamic simulation and behaviour analysis for online processing of live video from closed-circuit television (CCTV) system cameras. In this research, we focus on the last two components of the framework – the 3D visual scene simulator and the dynamic behaviour pattern recognizer, while the trajectory reconstruction and the other components of the framework are reported elsewhere [17][22]. In this paper, we will report the results of our experimentation with the model-driven behaviour pattern analyser, which works in pair with a 3D visual scene simulator as shown on Figure 1.

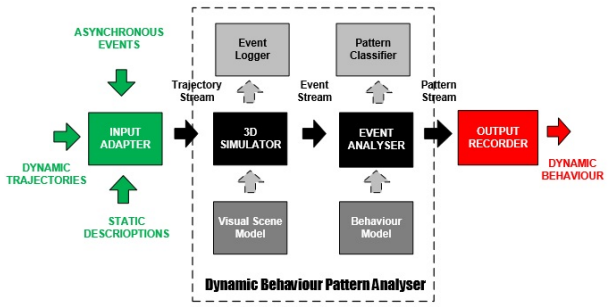

Figure 1. Data Flow in Dynamic Behaviour Analysis Framework

II. Dynamics Of The Visual Scene and Patterns of Dynamic Behaviour

The starting point for our analysis and the core of the entire framework for visual analytics is the ontology of the visual scene [16]. The purpose of this ontology is to provide an abstract representation of the information, which can be used in the logical analysis of the behaviour patterns. Various ontologies of bound worlds have been used for quite some time in Computer Science – i.e., in Computer Games [14] and Robotics [15]. Both areas share certain commonalities considering the fact that in both worlds the visual scene is observed from the point of view of a single "eye" (or pair of "eyes") – the "eye" of the robot or the "eye" of the gamer.

A. *Ontology of the visual scene*

Our ontology looks similar to the Spatio-Temporal Visual Ontology (STVO) presented in [21], although it has been developed completely independently on the base of the previous research of the authors in Artificial Intelligence (AI) and Computer Games.

At the top level of our ontology are the *Entities*, which are objects residing in the world. In Computer Games, objects are part of the game scene that can be managed or interacted with by the player. In Robotics, physical entities refer to objects that possess location in space and time that can be manipulated by robots. The objects recognized by a video camera can be specified implicitly by their physical attributes (location, velocity, orientation, etc.), which can be altered to execute some form of dynamic action. This is similar to the concept of "*Entity Manipulation*" element presented in the ontology of [14], where general Entities are classified on the basis of the actions they are capable of executing and their attributes. There are *Static Entities* that do not possess the ability to execute any action on their own; typically they are just part of the game world without changing their physical appearances. On the other side of the spectrum, there are *Dynamic Entities* possessing the ability to perform an action in order to manipulate the properties of other entities.

In Robotics, the ontologies contain an 'autonomous robot' agent that is capable of adapting to the changing environmental and executing actions on their own without human intervention [15]. The autonomous individual captured in the video footage may also be considered as a dynamic object capable of controlling its own movements and interaction with other objects on its own, without the need of intervention from any other objects. Individuals may form social groups in order to collaborate on achieving common goals. This is closely related to the definition of a 'robot group' in [15], where the term is specified as *"a group of robots organized to achieve at least one common goal"*. There is one special case of a group made out of only two individuals, which differs in being described by binary relations. In our ontology, it is classified as *pair* [16]. For example, if two paired individuals are talking to each other; they are also listening to each other during the conversation, while a third individual, observing the pair can only listen to them without talking to them.

The Game Ontology Project (GOP) introduced in [14] for describing and analysing games was built on the assumption formulated in [16] that the game elements and relationships between them are identified on the basis of visual perception and analysis of videogames. Without the insight of game designers' knowledge, their intentions or plans, the ontology is solely built on visual analysis of the game worlds. In other words, the ontology is based on how the authors perceived games as players and not necessarily as designers. Following this approach, from the visual observation of video footage, we can define the ontology of visual scene using the following core concepts:

Scene: provides information on boundaries of the space where objects are situated. It provides basis for coordinates of the restricted world monitored by physical video camera.

Object: an identified object that has physical location in space and time. There are three types of objects that can be identified: Static Objects, Dynamic Objects and Individuals.

Static Object: object that does not possess ability to execute any action and whose physical attributes can only be altered by dynamic objects or individuals. This type of object remains static for most of the time. Example: doors, shelves, stairs.

Dynamic Object: object that possess the ability to change physical properties of objects due to external factors or intervention or interaction of other objects at a particular time. Example: trolley, shopping product, envelope.

Individual: an autonomous dynamic object that has some degree of control over its movements. Individuals are capable of executing actions on their own without the need of intervention of other objects that may lead to interaction with other individuals or objects. Example: human, animal, autonomous robot.

Pair: two identified individuals that formed a relationship in which a certain degree of collaborative activities and interrelation can be observed between them. The activities in such a relationship can only be perceived as symmetric, anti-symmetric and generic types binary relations.

Group: an identified collection of three or more individuals exhibiting similar motions and potentially some level

of collaborative activities in order to achieve a mutual goal. A group can be treated as a single entity by aggregating all its participants' activities.

The above ontological concepts are the backbone of the 3D simulator of visual scene, which have been implemented as part of our framework using **jMonkeyEngine** [13]. The principles behind the 3D simulation have been introduced in our previous publication [16], while more details can be found in the PhD thesis of the first author [23]. The simulator has been extensively tested and shows excellent performance, matching the speed of video footage within the range 5-30 fps, which is sufficient for real-time applications.

B. Ontology of the dynamic patterns

The patterns of behaviour are derived from observation and analysis of the dynamics of objects previously identified in the visual scene. Assuming that we know the location of each individual, the position of their limbs relative to the body and the directions of movement and viewing at any moment of time, we can define a number of actions, which can be executed by those individuals. These actions are the building blocks of the complex patterns of dynamic behaviour. They can be recognized purely based on logical analysis, which is a cornerstone of our simulation approach.

The correlation between individual actions of the individuals and the events, which occur at the visual scene, can be modelled using three alternative ontological approaches:

- ***The actions are considered as changing the world and the events are only triggering them.*** In this approach, changes may or may not occur in time because the world remains in the same state if no activities are taking place. The changes are always caused by activities, while the events are relative to the time but independent from the actions. This approach is suitable for modelling actions that are instantaneous and triggered by events; the processes, unlike actions, have duration. It is commonly adopted in object-oriented modelling paradigm because the objects remain in the same state if no external activities are affecting them. This is the oldest approach widely employed in the early research in intelligent robots [15]. Similarity can also be found in the "Interface" conceptual element of the game ontology [14]. The input device provides the players means of sending signals to the game interface so that they can be turned into suitable actions. Whenever a player causes an event in the form of pressing a button, a corresponding action is executed on the screen. It may or may not change the state of the game world (change attributes of the entities of the game world). Time in this case can be completely disregarded as it does not influence the way events and actions occur. However, this approach leads to representational issues related to the so called "frame problem" in AI [18]. To tackle this problem, we have adopted the principle of inertia.
- ***The events are considered as changing the world and the actions are just collecting them.*** In this approach, the events are happening all the time, so the time is attributed to them. The state of the world in this case is defined in terms of the history of events. The world in such a case may or may not change depending on the events, not on the actions. The time measures the delay between events (frame update) but it does not initiate the changes. To that end, the actions would have to be defined through events as well. This approach is relatively new in Computer Science. It is less intuitive and leads to more complex logics [19]. But the effect of the events happening in the world according to this approach coincides with the effect of the actions, which changes in accordance with the previous approach if there is only one observer in the world, so in the case of a single camera this model is unnecessary complication.
- ***The world changes constantly with the time, the events and actions are just happening along the time line.*** In this approach, the changes are caused by the time while the actions are no longer instantaneous and have real physical duration. This approach has been successfully used in AI planning [20]. It would allow proper treatment of parallel activities, but may require synchronization of the visual data processing. This, in turn, would lead to a complicated implementation of multi-threaded services, which can run on a central server only.

The approach that has been adopted to model our world follows closely the first approach as outlined above. Our working assumption is that we have only one camera and all information collected from it is processed in a centralized manner. More complex approaches to the dynamic ontology could be introduced at a later stage, when considering multiple cameras monitoring the same scene. In that case the visual information processing will require synchronization in order to be analysed properly. This could involve several technical complications due to the need for synchronization of frame rates, elimination of overlapping signals, reducing the delay of frame updates, etc. If, for instance, the movements of one object are identified in one camera output but not in the others because of differences in their frame rate, discrepancies may occur between the data coming from two different cameras. This, in turn, may result in erroneous analytical output. A good candidate for adequate treatment in this case is the ontology of actions and time based on event structures.

Based on the combination of two ontologies outlined above, a language for describing the patterns of dynamic behaviour within the visual scene has been developed. Figure 2 presents the top-level class view of its ontology modelled using **Protégé**.

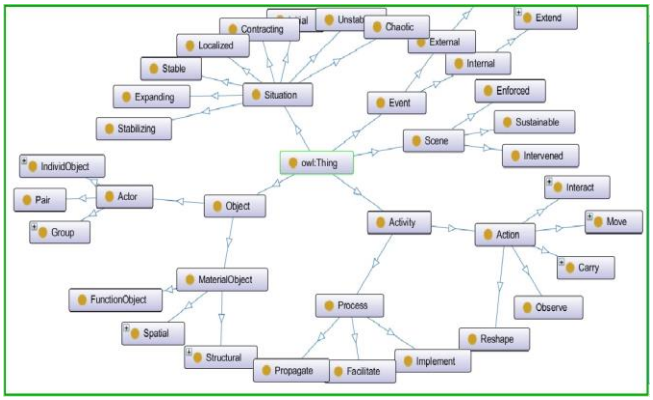

Figure 2. Ontology of the visual scene and dynamic behaviour

The language is the basis for implementation of the behaviour pattern analyser in our framework (Figure 1). All dynamic patterns are recognized on the basis of the syntactic analysis of the event logs, generated during the simulation in accordance with the syntax of this language [16]. More detailed specification of the language is provided in the PhD dissertation of the first author [23].

III. 3D Simulation Of Individual and Group Dynamics

The **3D Simulator** is a self-contained software application capable of simulating dynamic movements of individuals within the three-dimensional space of the visual scene in real-time, as illustrated in Figure 3. Using limited spatial information about the objects, such as approximate geometry, location and orientation, the simulator is capable to simulate the dynamics of the scene in real time. Thanks to the number of techniques borrowed from game programming, such as ray casting and ghosting [16], and by incorporating of a number of empirical laws of the naïve dynamics [31] the simulator is capable to perform simulation in real time with satisfactory performance.

The simulator possesses several features that are important for the subsequent analysis of the dynamic behaviour:

i) The rendering window, which allows observing the changes in a 3D scene visually.
ii) The parameters panel, which allows interactively adjusting the simulation parameters at runtime.
iii) Saving and loading of the simulation configuration defined in XML format.
iv) The console output panel, which allows tracing the events arising during the simulated scenario.
v) The generator of the event log defined in XML formatted files for subsequent analysis.

The graphical rendering of the scene is convenient for tuning and testing, but is not necessary for the analysis and can be switched off to improve the performance.

Figure 3 shows the three panels of the simulator – the visual output produced during the simulation, the event log generated by the logger and the parameter configuration menu for setting up the simulation parameters.

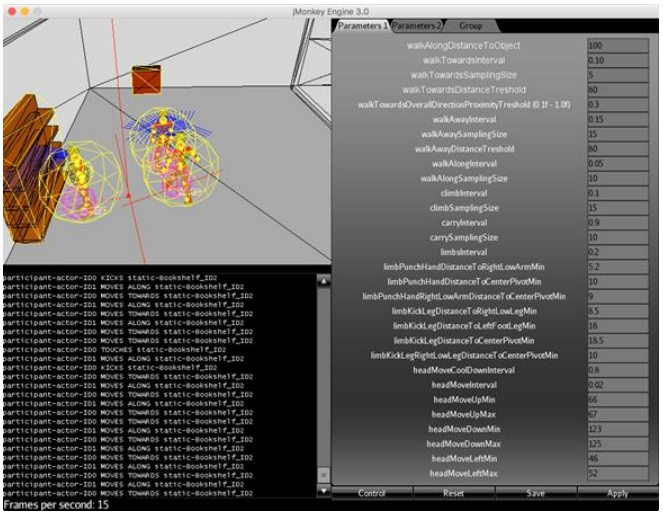

Figure 3. The simulator interface

All input data used by the simulator arrives at its input in real time in the form of a XML-formatted stream of data, created at the preliminary stage of visual processing of the original video footage by other components of the framework (Figure 1). It includes the following information:

- The reconstructed trajectories of moving objects describing their locations, orientation in space, direction of movement and viewing direction captured at specific time intervals.
- The physical properties of static entities residing in the visual scene extracted from the video footage offline or at run-time.
- The asynchronous events recognized in the video footage at the time of the analysis or the system events triggered by the simulator, such as changing the parameters of the simulation at run-time.

In addition to the 3D simulation, the simulator prepares the necessary data for behaviour analysis of the actual footage. It is doing this by generating a discrete log of the events occurring during the simulations. These events are detected and analysed by a separate module of the simulator using an event buffering technique commonly used in game programming. The detected events are logged as entries in the event log according to the syntax of the language, which describes the patterns of behaviour and analysed.

By replacing the continuous stream of purely physical data from the visual scene with discrete logs of the events it becomes possible to analyse the dynamic behaviour of the objects on the scene solely using symbolic methods.

IV. Individual and Group Dynamic Behaviour Analysis

The pattern analyser uses the output of the simulator for deriving the individual and group dynamic behaviour by processing the event log generated by the simulator. The event log, as the name implies, contains the relevant information about the events occurring during simulation.

From the 3D scene perspective, the events in most cases emerge from the detection of logical collisions between objects for which only partial data has been delivered to the simulator. The simulator itself generates the additional information needed for detecting the collisions. *The novelty of this approach is that the relations between the entities are established purely logically, based on the ontological model of the visual scene embedded in the simulation, rather than physically, based on the visual information extracted from the video footage*. In its current implementation the pattern analyzer module is capable of recognizing nearly 40 different patterns in real time (i.e., at a speed of up to 30fps). Amongst the more interesting patterns are:

- ✓ **"Somebody/a pair/a group is walking towards/away from something"**
- ✓ **"Somebody/a pair/a group is walking alongside something"**
- ✓ **"Somebody climbs on/off something"**
- ✓ **"Somebody goes up/down"**
- ✓ **"Somebody looks left/right/up/down"**
- ✓ **"Somebody drops something down"**
- ✓ **"Somebody holds something over something"**
- ✓ **"Somebody puts something on something"**
- ✓ **"Somebody picks up something from somewhere"**
- ✓ **"Somebody punches/kicks somebody else"**
- ✓ **"Somebody shakes hands with somebody else"**
- ✓ **"Two people form a pair"**
- ✓ **"Somebody joins/leaves a pair/a group"**
- ✓ **"A pair/group and another pair/group merge"**
- ✓ **"A group splits into a pair/group and another pair/group"**

The above patterns are described using a very small number of attributes – ***location*** of the body and its parts with ***position*** relative to the center of the body (for people), ***locations*** of the center and ***positions*** of its parts (for static objects), ***viewing directions*** and ***directions of movement*** (for moving objects). Despite their relative simplicity, these patterns can describe surprisingly rich set of complex behavioral patterns of interest in many applications.

In the current version of the analyzer, all patterns are purely relational in the sense that they incorporate a fixed number of parameters from specific type. In the next version, we are planning to introduce polymorphic parameters and inheritance, which would allow to account for the preliminary classification of static objects. This would increase the precision of simulation and would allow recognizing of more fine-grained patterns.

V. EXPERIMENTAL EVALUATION OF THE SIMULATOR AND THE PATTERN ANALYZER

Since the simulation is based on input data extracted from actual video footage, one of the problems we had to address in the experimental evaluation of the simulator and the analyzer was to acquire appropriate empirical data for conducting the experiments. In order to solve it, a simple keyboard-controlled emulator was implemented. It generates the synthetic data needed for the analysis directly from the "movies" produced using keyboard-controlled simulation. Because the speed of movement on the visual scene is relatively low, the dynamics of the generated "movies" is representative for the dynamics of the actual video footage so we can use the emulated data with satisfactory adequacy. Table I describes briefly some of the movies generated by this method, which have been used in the experiments. The data included in the files was used as a feed into the input during simulation of a given scenario for experimental validation and testing of the analyzer at runtime.

TABLE I. DESCRIPTION OF THE "MOVIE" FILES

File	Length	Scenario Description
004.xml	856 Frames	Two agents walking around the visual scene in a pair, then move away from each other, meet up and form a pair again.
005.xml	1128 Frames	Two agents walking around the visual scene; one of them climbs up the stairs
006.xml	808 Frames	A pair waiting. An agent walks in from a different room and moves towards it. He joins and the pair becomes a group.
007.xml	1479 Frames	Four agents walking around in two different rooms, separated by the wall. At one point they meet up in one of the rooms and form a group of four.
008.xml	1466 Frames	Three agents whose viewing directions change slowly; they form pairs and a group while walking around the visual scene.
009.xml	875 Frames	A crowd consisting of nine agents walking around, forming pairs, groups and browsing the premises, cluttering the visual scene for the entire time.
010.xml	857 Frames	Erratic movements of an agent whose viewing direction changes rapidly.

The accuracy of the analyser was estimated through replaying the "movies" recording different scenarios as described in Table I and comparing the logs produced by the analyser with the actual content of the "movies". During these experiments, the pattern analyser was operating in parallel with the simulator and was reporting the exact movie frame at which the corresponding pattern was recognized. To verify the patterns, we compared the actual changes in the agent properties and the predicted changes of these properties over several visual frames, which delimit the boundaries of a specific time period.

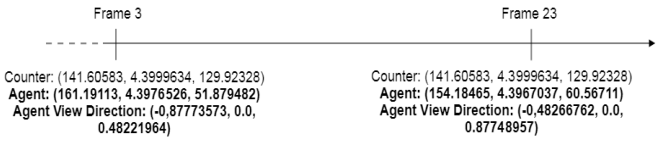

Figure 4. Changes of spatial properties of an Agent over time.

Figure 4 depicts such a timeline showing the changing spatial properties of an individual agent. While the static object (the shop counter in this case) remained in the same location, the position of the agent and its orientation changed over the sequence of 20 frames. At the end, the agent not only came closer to the counter but also changed its direction of movement, pointing towards it. In order to recognize that the agent started walking towards the counter, an additional ray casting was performed to detect if any other entities

(static or dynamic) are not between them. This is necessary to avoid situations when patterns are being reported despite the fact that potential obstacles may be located between the entities involved, such as tills, shelves or walls. By parameterizing the set of rules for capturing a given pattern the configuration of the simulator can be also adjusted to fulfil system requirements in real-time.

A similar experimental setup was used to test the pattern analyzer. For this purpose, each frame was timestamped in the movie file, which was generated during the simulation. By comparing the timestamps of the frame at which the pattern can be identified during the recording phase with the timestamp of the frame at which the same pattern has been recognized during the analysis phase we can calculate the delay in recognition of the patterns. Table II presents the delays in reporting the recognition of several dynamic patterns while replaying the movies at 30 fps. It is obvious that the analyzer is efficient and the delay is not substantial.

TABLE II. DELAY DUE TO COMPUTATIONAL AND RENDERING PROCESS OF THE MOVIE FILES

Pattern	**Critical Frame**	**Delay**
"Walking towards something"	23	0.44%
"Walking towards something while in a group"	45	0.39%
"Walking away from something"	105	0.31%
"Walking along something while agent is in a group"	133	0.27%
"Climbing something up"	214	0.22%
"Forming a pair"	442	0.13%
"Forming a group"	663	0.72%
"Group moving towards something"	747	4.51%
"Group moving along something"	1013	5.81%
"Leaving a group"	211	0.14%

We have also extensively tested the pattern analyser by varying the speed of recording and the speed of replaying. The results of the experiments are satisfactory, but the limited space of this paper does not allow reporting all of them.

Further series of tests were conducted to estimate the degree to which the pattern analyser is immune to degradation of computational resources. For this purpose, we forced the analyser to skip frames and estimated the delay in reporting the recognized patterns at different speed of replaying. Table III presents the delay in recognition dependent on the frame skipping rate at 30fps speed. Again, the results are very encouraging and prove the feasibility of the model-driven simulation-based methodology of analysis.

TABLE III. DELAY DUE TO COMPUTATIONAL AND RENDERING PROCESS OF THE MOVIE FILES

Pattern	**Critical Frame**	**Skipped frames**	**Delay**
"Walking towards something"	23	50%	0.22%
		66%	0.38%
		76%	0.50%
		83%	0.41%
		90%	0.41%
"Walking towards something while in a group"	45	50%	0.45%
		66%	1.82%
		76%	8.37%
		83%	7.59%
		90%	9.4%
"Walking away from something"	105	50%	0.26%
		66%	0.38%
		76%	0.42%
		83%	0.35%
		90%	0.37%

VI. CONCLUSION

In this paper, we have presented the results of an experimental analysis of a 3D simulator and the associated model-driven analyser, which are parts of a framework for individual and group dynamic behaviour analysis in video surveillance. The results convincingly demonstrate the feasibility of this approach to the analysis and build the necessary confidence in the possibility to use model-driven and simulation-based approach in video analytics with a wide range of potential applicability in video surveillance. During the next phase of research we plan to extend the simulator with the possibility to model the shapes of the static objects on the scene, to account the physical boundaries of the space and to make use of the sight sense of the agents, which would allow to analyse more precisely the behaviour and to recognize more complex patterns.

ACKNOWLEDGMENT

This research is sponsored by The Vinyl Factory Limited - London, United Kingdom.

REFERENCES

[1] C. Hu and S. Wo, "An efficient method of human behavior recognition in smart environments", In Int. Conf. on Comp. Application and System Modeling, Vol. 12, pp. 690–693, 2010.

[2] K. Yordanova, "Modelling Human Behaviour Using Partial Order Planning Based on Atomic Action Templates", In 7th Int. Conf. on Intelligent Environments, pp. 338–341, 2011.

[3] C. Wang and F. Wang, "A Knowledge-Based Strategy for Object Recognition and Reconstruction", In Int. Conf. on Information Technology and Computer Science, pp. 387–391, 2009.

[4] M. Attamimi, T. Nakamura, and T. Nagai, "Hierarchical multilevel object recognition using Markov model," In 21st IEEE International Conference on Pattern Recognition, pp. 2963-2966, November, 2012.

[5] S. Wu, B. Moore, and M. Shah, "Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes", In Proc. IEEE Conf. on Computer Vision and Pattern Recognition CVPR2010, pp. 2054–2060, 2010.

[6] P. Saboia and S. Goldenstein, "Crowd Simulation: Improving Pedestrians' Dynamics by the Application of Lattice-Gas Concepts to the Social Force Model", In 24th SIBGRAPI Conf. on Graphics, Patterns and Images (Sibgrapi), pp. 41–47, 2010.

[7] R. Guo and H. Huang, "A mobile lattice gas model for simulating pedestrian evacuation", In Physica, Part A: Stat. Mechanics and its Applications, Vol. 387, pp. 580–586, 2007.

[8] 3VR Inc., 3VRVideoIntelligence Platform, 2015 [http://3vr.com/products/videoanalytics; last access 06/05/2018].

[9] Agent Video Intelligence Ltd., "savVi Real-Time Event Detection",

2016 [https://www.agentvi.com/products/; last access 06/05/2018].

[10] PureTech Systems Inc., "Video Analytics", 2015 [http:// www.puretech systems.com/video-analytics.html; last access 06/05/2018].

[11] IndigoVision, Control Center, IndigoVision's Security Management Solution, 2017 [http://www.indigovision.com/products/management-software/; last access 06/05/2018].

[12] IBM, *Intelligent Video Analytics*, 2017 [http://www.ibm. com/uk-en/marketplace/video-analytics-for-security; last access 06/05/2018].

[13] R. Kustener, *JMonkeyEngine 3.0 Beginner's Guide*, Birmingham: Packt Publ., 2013.

[14] J.P. Zagal, M. Mateas, C. Fernández-Vara, B. Hochhalter, and N. Lichti, "Towards an ontolqgical language for game analysis", In International Perspectives on Digital Games Research, 21, p.21, 2007.

[15] IEEE Std. 1872-2015 1–60, *Standard Ontologies for Robotics and Automation*, IEEE, 2015.

[16] P. Gasiorowski, V. Vassilev and K. Ouazzane, "Simulation-based Visual Analysis of Individual and Group Dynamic Behavior", In: Proc. 20th Int. Conf. Image Processing, Computer Vision & Pattern Recognition (IPCV'16), CSREA Press, pp. 303-309, 2016.

[17] M. Afzal, K. Ouazzane, V. Vassilev and Y. Patel, "Incremental Reconstruction of Moving Object Trajectory", In Proc. Of The First International Conference on Applications and Systems of Visual Paradigms, pp. 24–29, 2016.

[18] M. Shanahan and M. Witkowski,, "High-Level Robot Control through Logic", In Castelfranchi, C., Lespérance, Y. (Eds.), Intelligent Agents VII - Agent Theories, Architectures and Languages, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 10.1007, pp. 104–121, 2000.

[19] R. Kowalski and M. Sergot. A Logic-based Calculus of Events. New Generation Computing 4: 67–95, 1986.

[20] J. F. Allen, Maintaining knowledge about temporal intervals, Communications of the ACM, 26(11): 832–843, 1983.

[21] J.I. Olszewska "Spatio-Temporal Visual Ontology", School of Computing and Engineering University of Huddersfield, UK 1st EPSRC/BMVA Workshop on Vision and Language, Brighton, UK, September 2011.

[22] M. Afzal, K. Ouazzane and V. Vassilev "K-Nearest Neighbours-Based Classifiers for Moving Objects Trajectory Reconstruction", The Third International Conference on Applications and Systems of Visual Paradigms, Venice, June 2018.

[23] P. Gasiorowski, "Individual and group dynamic behaviour patterns in bound spaces", PhD Thesis, London Metropolitan University, 2017 [http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.740053].