VISUAL 2018

The Third International Conference on Applications and Systems of Visual Paradigms

ISBN: 978-1-61208-647-7

June 24 - 28, 2018

Venice, Italy

VISUAL 2018 Editors

Petre DINI, IARIA, USA

VISUAL 2018

Forward

The Third International Conference on Applications and Systems of Visual Paradigms (VISUAL 2018), held between June 24, 2018 and June 28, 2018 in Venice, Italy, continued the inaugural event in putting together complementary domains where visual approaches are considered in a synergetic view.

Visual paradigms were developed on the basis of understanding the brain's and eye's functions. They spread over computation, environment representation, autonomous devices, data presentation, and software/hardware approaches. The advent of Big Data, high speed images/camera, complexity and ubiquity of applications and services raises several requests on integrating visual-based solutions in cross-domain applications.

We take here the opportunity to warmly thank all the members of the VISUAL 2018 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated their time and effort to contribute to VISUAL 2018. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also gratefully thank the members of the VISUAL 2018 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that VISUAL 2018 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of applications and systems of visual paradigms. We also hope that Venice, Italy provided a pleasant environment during the conference and everyone saved some time to enjoy the unique charm of the city.

VISUAL 2018 Chairs

VISUAL Steering Committee

Vijayan K. Asari, University of Dayton, USA
Robert S. Laramee, Swansea University, UK
Luciano Pereira Soares, Insper, Brazil
Jibonananda Sanyal, Oak Ridge National Laboratory | Mississippi State University, USA
Irene Yu-Hua Gu, Chalmers University of Technology, Sweden

VISUAL Industry/Research Advisory Committee

Kresimir Matkovic, VRVis Research Center, Vienna, Austria
Jinrong Xie, eBay Inc., USA
Afzal Godil, National Institute of Standards and Technology, Gaithersburg, USA

Silvia Biasotti, Consiglio Nazionale delle Ricerche, Genova, Italy
Carolin Helbig, Helmholtz Centre for Environmental Research (UFZ), Germany

VISUAL 2018

Committee

VISUAL Steering Committee

Vijayan K. Asari, University of Dayton, USA
Robert S. Laramee, Swansea University, UK
Luciano Pereira Soares, Insper, Brazil
Jibonananda Sanyal, Oak Ridge National Laboratory | Mississippi State University, USA
Irene Yu-Hua Gu, Chalmers University of Technology, Sweden

VISUAL Industry/Research Advisory Committee

Kresimir Matkovic, VRVis Research Center, Vienna, Austria
Jinrong Xie, eBay Inc., USA
Afzal Godil, National Institute of Standards and Technology, Gaithersburg, USA
Silvia Biasotti, Consiglio Nazionale delle Ricerche, Genova, Italy
Carolin Helbig, Helmholtz Centre for Environmental Research (UFZ), Germany

VISUAL 2018 Technical Program Committee

Amr Abdel-Dayem, Laurentian University, Canada
Andrew Adamatzky, University of the West of England, Bristol, UK
Athos Agapiou, Eratosthenes Research Centre / Cyprus University of Technology, Cyprus
Konstantin Aksyonov, Ural Federal University, Russia
Leyteris Anastasovitis, Information Technologies Institute - Centre for Research and Technology Hellas (ITI-CERTH) / University of Macedonia, Greece
Djamila Aouada, University of Luxembourg, Luxembourg
Vijayan K. Asari, University of Dayton, USA
George Baciu, The Hong Kong Polytechnic University, Hong Kong
Yufang Bao, Fayetteville State University, USA
Ardhendu Behera, Edge Hill University, UK
Saeid Belkasim, Georgia State University, Atlanta, USA
Yannick Benezeth, Univ. Bourgogne Franche-Comté, France
Stefano Berretti, University of Florence, Italy
Silvia Biasotti, Consiglio Nazionale delle Ricerche, Genova, Italy
Hans-Peter Bischof, Rochester Institute of Technology, USA
Kadi Bouatouch, IRISA / Unviversity of REnnes 1, France
Vittoria Bruni, University of Rome "La Sapienza", Italy
Kevin Chalmers, Edinburgh Napier University, UK
Satish Chand, Jawaharlal Nehru University, India
Bin Chen, Purdue University Northwest, Hammond, USA
Haeyong Chung, University of Alabama in Huntsville, USA
Sara Colantonio, Institute of Information Science and Technologies - ISTI | National Research Council of

Italy – CNR, Pisa, Italy
José Mario De Martino, University of Campinas, Brazil
Saverio Debernardis, Polytechnic Institute of Bari, Italy
Cosimo Distante, Institute of Applied Sciences and Intelligent Systems “ScienceApp" | Consiglio Nazionale delle Ricerche, Italy
Jana Dittmann, Otto-von-Guericke-University Magdeburg, Germany
Anastasios Doulamis, National Technical University of Athens, Greece
Pierre Drap, Aix-Marseille University, France
Soumya Dutta, Ohio State University, USA
Jorge Fernández Berni, Instituto de Microelectrónica de Sevilla | Universidad de Sevilla-CSIC, Spain
Zlatko Franjcic, Chalmers University of Technology, Gothenburg, Sweden
Emanuele Frontoni, Università Politecnica delle Marche, Ancona, Italy
Francesco Gabellone, National Research Council (CNR) - Istituto per i Beni Archeologici e Monumentali (IBAM), Lecce, Italy
Daniela Giorgi, Institute of Infomation Science and Technologies (ISTI) - National Research Council of Italy (CNR), Pisa, Italy
Afzal Godil, National Institute of Standards and Technology, Gaithersburg, USA
Manuel González-Hidalgo, University of the Balearic Islands, Spain
Prashant Goswami, Blekinge Institute of Technology, Karlskrona, Sweden
Valerie Gouet-Brunet, IGN, France
Denis Gracanin, Virginia Tech, USA
Sebastian Grottel, TU Dresden, Germany
Irene Yu-Hua Gu, Chalmers University of Technology, Sweden
Kun Guo, School of Psychology | University of Lincoln, UK
Carolin Helbig, Helmholtz Centre for Environmental Research (UFZ), Germany
Olaf Hellwich, Technische Universität Berlin, Germany
Xiang Huang, Argonne National Laboratory, USA
Laura Igual, Universitat de Barcelona, Spain
Jiri Jan, Brno University of Technology, Czech Republic
Mark W. Jones, Swansea University, UK
Hamid Reza Karimi, Politecnico di Milano, Italy
Santosh KC, University of South Dakota (USD), USA
Andreas Koschan, University of Tennessee, USA
Dimitrios Koukopoulos, University of Patras, Greece
Gauthier Lafruit, Brussels University, Belgium
Robert S. Laramee, Swansea University, UK
José L. Lázaro-Galilea, University of Alcalá, Spain
Quan Li, Hong Kong University of Science and Technology, Hong Kong
Ruming Li, Chongqing University, China
Kari Lilja, Satakunta University of Applied Sciences | University Consortion Pori / Tampere University of Technology, Finland
Lars Linsen, Jacobs University, Germany
Massimiliano Lo Turco, Politecnico di Torino, Italy
Kresimir Matkovic, VRVis Research Center, Vienna, Austria
Cyrille Migniot, Université Bourgogne Franche-Comté, France
Thomas Moeslund, Aalborg University, Denmark
Bartolomeo Montrucchio, Politecnico di Torino, Italy
Sander Münster, Technische Universität Dresden, Germany

Copyright Information

Table of Contents

3D Simulation-based Analysis of Individual and Group Dynamic Behaviour in Video Surveillance 1
Pawel Gasiorowski, Vassil Vassilev, and Karim Ouazzane

K-Nearest Neighbours Based Classifiers for Moving Object Trajectories Reconstruction 7
Muhammad Afzal, Karim Ouazzane, and Vassil Vassilev

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.

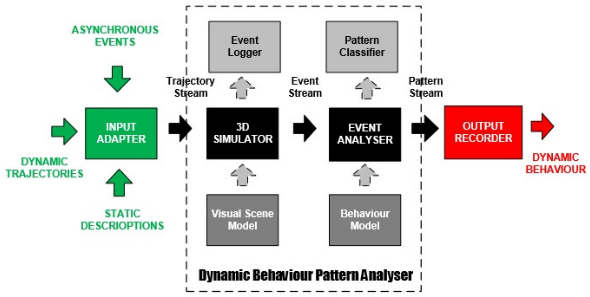

Figure 1. Data Flow in Dynamic Behaviour Analysis Framework

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

 ISBN: 978-1-61208-647-7

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**.

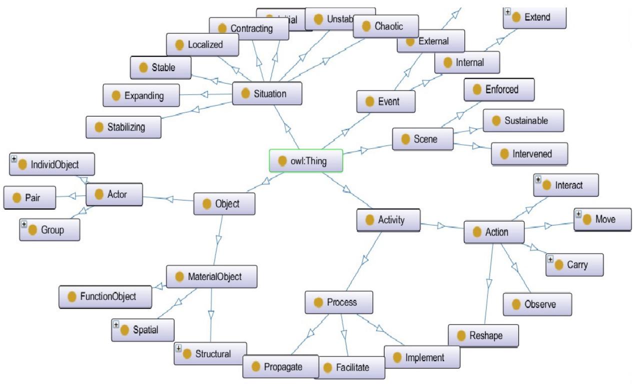

Figure 2. Ontology of the visual scene and dynamic behaviour

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 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.

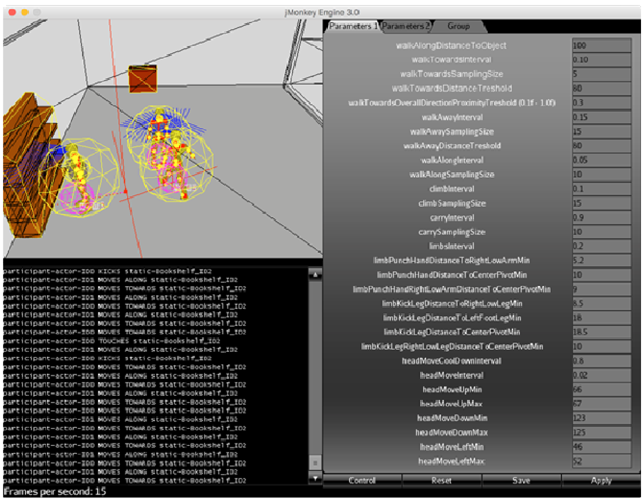

Figure 3. The simulator interface

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 along 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 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, used as a feed into the input during simulation of a given scenario for experimental validation 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
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.

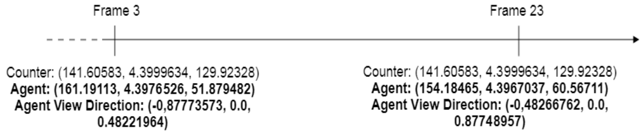

Figure 4. Changes of spatial properties of an Agent over time.

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. We have also extensively tested the pattern analyser by varying the speed of recording and the speed of replaying with satisfactory results. Further series of tests were conducted to estimate the degree to which the 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 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 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%

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, London.

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. 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. 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, Lect. Notes Comp. Sc., 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 Work. 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].

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.

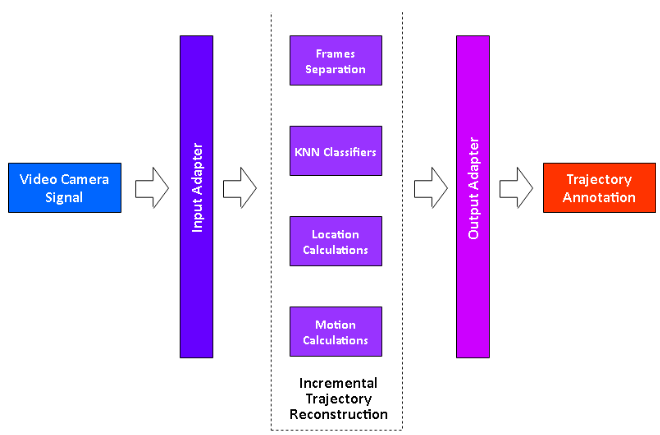

Figure 1. Incremental trajectory reconstruction using KNN classifiers

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.

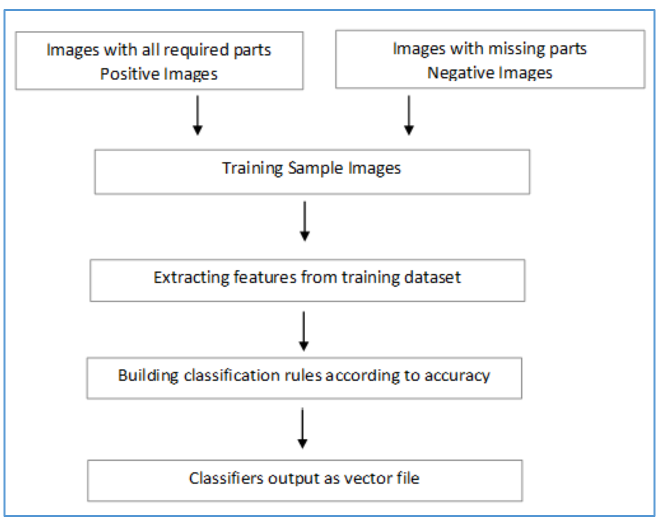

Figure 2. Classifiers learning steps

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:

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

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

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

while the j^{st} feature vector is

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

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

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

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:

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

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

$$s_j = [\, s_{1,j}; \ldots; s_{ij}; \ldots\ s_{m,j}] \in \Re^{m+1} \tag{9}$$

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

$$S = [s^1; \ldots, s^j; \ldots; s^m\,] \in \Re^{m+m} \tag{11}$$

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:

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

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

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

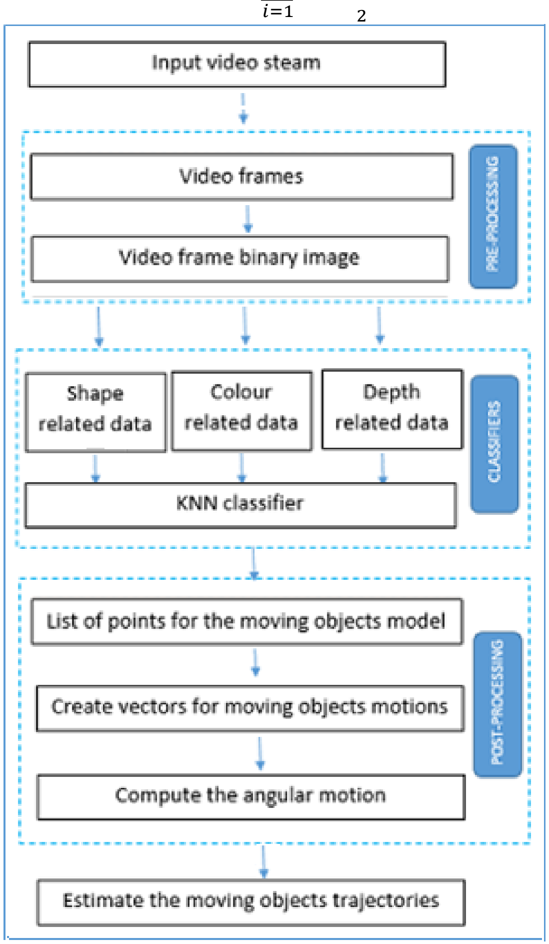

Figure 3. Flow of video stream analyzed using KNN classifiers

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.

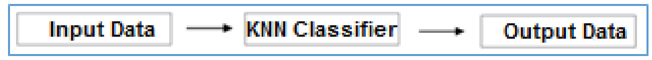

Figure 4. Classifiers execution and extracting information steps

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.

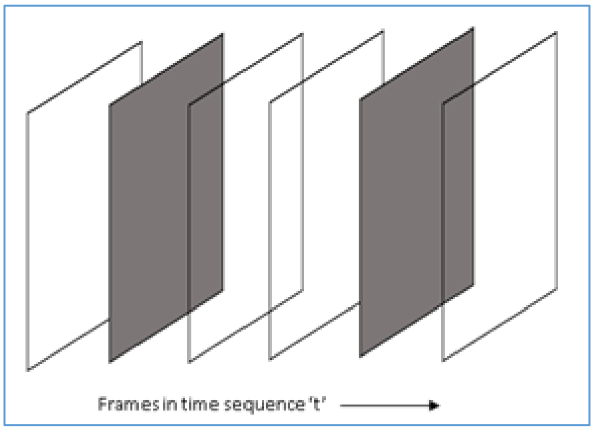

Figure 5. Sequence of frames

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.

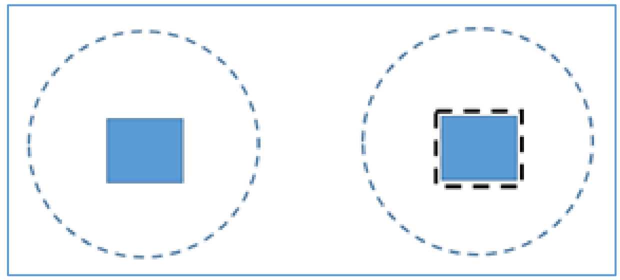

Figure 6. Shaping its projections on the frame using classifiers

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***
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**

 ISBN: 978-1-61208-647-7

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.

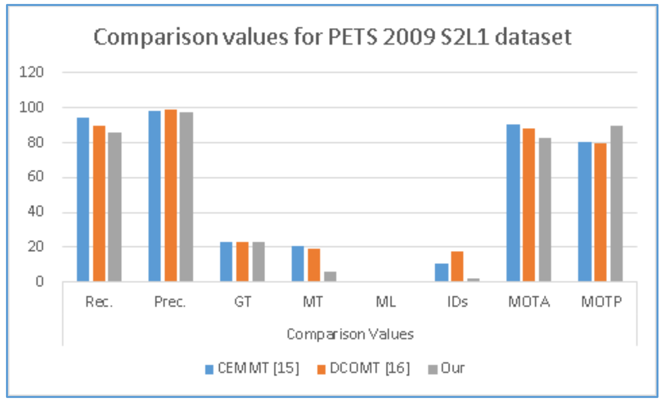

Figure 7. Comparison of the values for PETS 2009 S2L1 dataset (Y-Axis is showing the percentage)

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.

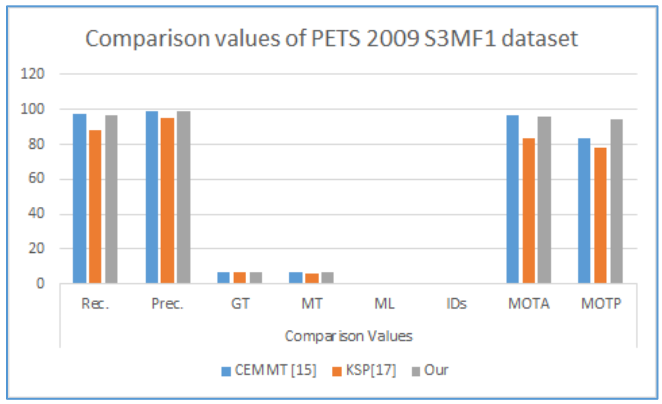

Figure 8. Comparison of the values for PETS 2009 S3MF1 dataset (Y-Axis is showing the percentage)

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.

 ISBN: 978-1-61208-647-7

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.

 ISBN: 978-1-61208-647-7