OBJECT-ORIENTED LOGIC PROGRAMMING OF 3D INTELLIGENT VIDEO SURVEILLANCE SYSTEMS: THE PROBLEM STATEMENT

[1,2]Alexei A. Morozov, [1]Olga S. Sushkova, [1]Alexander F. Polupanov

[1]Kotel'nikov Institute of Radioengineering and Electronics of RAS, http://www.cplire.ru
Moscow 125009, Russian Federation
[2]Moscow State University of Psychology & Education, http://www.mgppu.ru
Moscow 107045, Russian Federation

morozov@cplire.ru, o.sushkova@mail.ru, sashap55@mail.ru

Abstract. **An approach to the 3D intelligent video surveillance based on the means of the object-oriented logic programming is proposed. In contrast to the conventional 2D video surveillance, the methods of 3D vision provide reliable recognition of parts of the human body that makes possible a new statement of the problem and efficient practical application of methods of people behaviour analysis in the video surveillance systems. The logic-based approach to the intelligent video surveillance allows easy definition of people complex behaviour in terms of simpler activities and postures. The goal of this work is to implement the advantages of the logic programming approach in the area of 3D intelligent video surveillance.**

Keywords: **: intelligent video surveillance; 3D vision; object-oriented logic programming; Actor Prolog; Kinect; skeleton; complex event recognition; machine vision; technical vision; Prolog to Java translation**

UDC 681.3.06

Bibliography – 61 references *Received 29.11.2017*

RENSIT, 2017, 9(2):205-214 **DOI: 10.17725/rensit.2017.09.205**

Contents

1. INTRODUCTION

The 3D video surveillance term has different meanings in research papers:

1. Initially, this term was associated with advanced video surveillance systems supporting real-time 3D visualization of indoor environment, buildings, and people to be monitored, as well as application of virtual and augmented reality technologies in the video surveillance [1]–[4].
2. This term could be associated also with so-called 3D approaches to motion recovery from 2D images [5].
1. Nowadays, this term could be associated with using 3D/2.5D data acquisition equipment [6] like time-of-flight (ToF) cameras, structured light sensors, stereo cameras, laser scanners (LiDARs) [7], and Flash Ladars [8] in video surveillance.

Availability of 3D data acquisition equipment leads to considerable progress in video surveillance related research areas such as background subtraction and tracking people [9], [10], person identification [11], [12], face recognition [13], posture recognition [14], [15], gesture recognition [16], people counting [17], human activity recognition [18], etc. because the use of the 3D/2.5D data makes possible to solve many complicated problems of computer vision caused by the presence of shadows, variations in illumination, view-point changes, cluttered background, and occlusions. And a more important point is in that this technological advance enables progress in solving complex intelligent video surveillance problems [19] like human-human interaction recognition and prediction [20], abnormal event detection for healthcare systems [21]-[23], unusual events detection [24], the behaviour of customers [25], monitoring of people with disabilities [26], [27], aggressive behaviour and anger detection [28], gait recognition [29] and biometric surveillance [30], and analyzing the human motion to aid clinical decision making [31], [32].

In contrast to the conventional 2D video surveillance, the methods of 3D vision provide reliable recognition of the parts of the human body that makes possible a new statement of the problem and efficient practical application of methods of people behaviour analysis in the video surveillance systems. According to [33], people behaviour is a kind of activity which is a response to some stimuli (internal, external, conscious, or unconscious). For example, such interactions of people as a friendly handshake, a fight, the stealing of a suitcase from another person, etc. could be forms of behaviour. Analysis of the behaviour requires information about the context and other factors influencing it [33]. This information can include video scene geometry, place and presence of objects, time constraints, etc. Thus, the availability of exact information about the co-ordinates of the parts of the body and scene geometry provided by means of the 3D vision is a key to the automation of people behaviour analysis, recognition, and understanding.

Mathematical logic and logic programming were recognized as convenient means for video scene context description and analysis. The idea of the logic programming based approach to the people behaviour analysis is in usage of logical rules for description and analysis of people activities. Knowledge about the human body structure, object properties, time constraints, and scene geometry is encoded in the form of the rules in a logic language and is applied to the output data of some low-level procedures of detection of people and objects.

This idea was developed in the framework of the conventional 2D intelligent video surveillance, and there are a lot of studies based on it. The usage of temporal logic programs in the W^4 real-time system for representation of actions and analysis of people movements was reported in [34]. The VidMAP video surveillance system that combines real-time computer vision algorithms with the Prolog-based logic programming had been announced in [35]. Later this research group has proposed an extension of the predicate logic with the bilattice formalism that permits processing of uncertainty in the reasoning about the video scenes [36]. The VERSA general purpose framework for defining and recognizing events in live or recorded surveillance video streams was described in [37]. The VERSA system is based on SWI-Prolog. A real-time complex audio-video event detection based on the Answer Set Programming approach was proposed in [38]. Heuristic rules based on a temporal logic are used for describing semantics of video scenes in [39]. A system [40] was designed for recognition of so-called long-term activities (such as "fight" or "meeting") consisting of sequences of short-term activities (such as "walking", "running", "active movement on the spot", "standstill", and "sharp movements") using a logic programming implementation of the Event Calculus. Later this approach was extended to handle the uncertainty that occurs in human activity recognition [41]. The ProbLog probabilistic logic language was used for this purpose. A Prolog-based video surveillance framework for robust detecting abandoned objects was described in [42]. It is believed that the system automatically identifies interrelations between the observed people in order to draw a conclusion about who exactly owns the observed objects, and thus reduce the probability of a false alarm.

Curiously there are not so many papers that utilize logic programming for 3D video data processing. The method of recognition of long-term activities [40] was adapted [26] for processing the data collected with the help of Kinect. The monitoring system is intended for the analysis of the behaviour of a person in a support room. As reported in [26], the skeleton data were not used for the analysis of long-term activities though the author calculates so-called Activity Level coefficient on the base of the average displacement of all tracked and inferred joints of the skeleton. The probabilistic approach to the long-term activity recognition [41] was adapted to the development of a distributed monitoring system [43] that provides health-care assistance in a smarthome setting. A reasoning engine based on the Description Logics (DL) formalism was used for Kinect-based posture and gesture recognition [44]. Prolog rules were used in the RoboSherlock system [45] to distinguish between actions such as "take", "put", "fill", "push". The logical inference takes into account context information about the properties of the objects. For example, to recognize the "fill" action, it is necessary to know that the object manipulated by a person is a container, food, or an ingredient [45].

The logic-based approach to the intelligent video surveillance has the following obvious advantages [26]:

1. One can easily incorporate domain knowledge into the recognition process.
2. Reasoning about the behaviour of people can be very difficult, but for the developer of the system of intelligent video surveillance, it always remains fundamentally understandable.
3. One can easily define a complex activity in terms of simpler activities.

Our goal is to implement these advantages of the logic approach in the area of the 3D intelligent video surveillance by means of the object-oriented logic programming.

2. THE OBJECT-ORIENTED LOGIC PROGRAMMING OF THE INTELLIGENT VIDEO SURVEILLANCE SYSTEMS

The method of the object-oriented logic programming of the intelligent video surveillance systems was developed in [46]-[52]. The distinctive features of the method are usage of the Actor Prolog object-oriented logic language [53]-[55] and translation of the logic programs of intelligent video surveillance to the Java language [47], [56]. In Actor Prolog the text of a logic program consists of separate classes [53]. Parallel processes represent a kind of instances of classes [54]. The object-oriented means of the logic language allow one to split a program to interacting parallel processes that implement various stages of image processing and scene analysis while translation to the Java language provides a high performance sufficient for analyzing real-time video, as well as reliability and stability of the work of the intelligent video surveillance software.

A conventional approach to recognition of human activity includes low-level and high-level stages of the video processing. In the case of the 2D video surveillance, the low-level stage includes background subtraction of a video image, detection of people and cars in the video image, construction of the trajectories of moving objects, evaluation of the velocities of the objects, etc. The high-level stage includes logical analysis of graphs of the trajectories that are supplied by the low-level analysis algorithms and displaying the results on the screen [50], [51]. The low-level stage of the video processing is implemented in Java [57] in special built-in classes of Actor Prolog.

Since the 2D video surveillance cannot provide a stable recognition of human body parts in most real-world applications, the graphs are used for describing trajectories of blobs (separated areas of a foreground image). The graphs include information about the co-ordinates and the velocities of the blobs, as well as a set of statistical metrics describing the regularity of the motion [48]. The blob motion statistics is collected by the low-level analysis procedures and is used for discrimination of running people, people riding bicycles, and cars in the video scene. Usually simple fuzzy inference is implemented for this purpose on the base of the statistical metrics and standard arithmetical predicates of the logic language.

In the case of the 3D video surveillance, the goals of the low-level video analysis are essentially different. For instance, the standard software of the Kinect 2 time-of-flight camera can provide a ready for use foreground blobs, as well as up to six skeletons of people to be monitored [6], [18], [19], [58]-[61]. Thus, the minimal pre-processing of the data can include just a conversion of the images of the skeletons into the terms of the logic language. The high-level stage of the video data processing can be implemented in accordance with the former principles but the information about the video scene is essentially more complete, that gives an opportunity to infer much more of logical consequences on the semantics of the video scene.

3. AN EXAMPLE OF LOGICAL INFERENCE ON 3D DATA

Let us consider a simple logic program that analyses 3D data collected using the Kinect 2 ToF camera.

The data include depth maps and images of skeletons of people, but the program has to analyze the images of the skeletons only and the depth maps are to be used only for the visualization of the video scene. Each skeleton contains 25 joints of various statuses (See **Fig. 1**). The statuses of joints of skeleton can be the following:

1. Tracked – the corresponding part of the body is directly observed by the camera.
2. Inferred – the position of the joint is a hypothetical one.
3. Unknown – the device has no information about this joint.

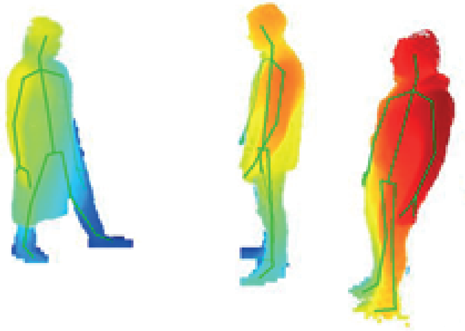

Fig. 1. *The images of skeletons and depth maps are the input data of the logic program. The depth maps are indicated by colours.*

Thus, the following set of domains (data types) can be defined in Actor Prolog to describe the data structures to be processed:

```
DOMAINS:
Skeletons = Skeleton*.
```

The `Skeletons` type is a list of elements of the `Skeleton` type.

```
Skeleton = {
         identifier: INTEGER,
         head: SkeletonJoint,
         spine: SkeletonSpine,
         left_arm: SkeletonArm,
         right_arm: SkeletonArm,
         left_leg: SkeletonLeg,
         right_leg: SkeletonLeg
         }.
```

The `Skeleton` type is an underdetermined set [53] of Actor Prolog. The names of the elements of the skeleton as well as the names of corresponding types are obvious:

```
SkeletonSpine = {
         neck: SkeletonJoint,
         shoulder: SkeletonJoint,
         mid: SkeletonJoint,
         base: SkeletonJoint
         }.
SkeletonArm = {
         shoulder: SkeletonJoint,
         elbow: SkeletonJoint,
         wrist: SkeletonJoint,
         hand: SkeletonJoint,
         tip: SkeletonJoint,
         thumb: SkeletonJoint
         }.
```

```
SkeletonLeg = {
         hip: SkeletonJoint,
         knee: SkeletonJoint,
         ankle: SkeletonJoint,
         foot: SkeletonJoint
         }.
SkeletonJoint = {
         status: SkeletonJointStatus,
         position: VertexPosition3D,
         orientation: JointOrientation3D
         }.
```

The status of a joint is a symbol. Each joint is characterized by a position in 3D space, i.e. by three numbers, by orientation, i.e. by four numbers, and by two pairs of 2D co-ordinates that correspond to this joint and are obtained from the infrared and colour images:

```
SkeletonJointStatus =
        ’TRACKED’; ’INFERRED’; ’UNKNOWN’.
VertexPosition3D = {
         point: Point3D,
         velocity: Point3D,
         acceleration: Point3D,
         mapping1: Point2D,
         mapping2: Point2D
         }.
JointOrientation3D = q(
         Numerical,
         Numerical,
         Numerical,
         Numerical).
Point3D = p(Numerical,Numerical,Numerical).
Point2D = p(Numerical,Numerical).
```

Suppose that images of skeletons are transferred from the device in real time. Let us declare a set of predicates analyzing the 3D data:

```
PREDICATES:
imperative:
analyse_skeletons(Skeletons) - (i);
nondeterm:
a_hand_of_a_person
     (Skeletons,INTEGER,VertexPosition3D)
          - (i,o,o);
determ:
is_near_a_head_of_a_colleague
     (Skeletons,INTEGER,VertexPosition3D)
         - (i,i,i);
```

The analyse_skeletons predicate checks whether there are at least two persons in the video scene and one person punches in the face another person. If there are such persons, the predicate outputs a message in the screen (see **Fig. 2**). Here is the definition of the predicate:

```
CLAUSES:
analyse_skeletons(S):-
   a_hand_of_a_person(S,Id,P),
   is_near_a_hand_of_a_colleague(S,Id,P),!,
   graphic_window ? draw_text(
      0.93,0.08,
      "A punch in the face is detected!").
analyse_skeletons(_).
```

The a_hand_of_a_person predicate is nondeterministic. This predicate returns co-ordinates of hands of persons recognized in the scene. Note, that in the Actor Prolog language, the == operator corresponds to the = ordinary unification of the standard Prolog.

```
a_hand_of_a_person([S|_],Id,P):-
     S == {
          identifier:Id,
          left_arm:{
               hand:{
                  status:’TRACKED’,
                  position:P|_}
          |_}
     |_}.
a_hand_of_a_person
          ([S|_],Id,P):-
     S == {
          identifier:Id,
          right_arm:{
              hand:{
                  status:’TRACKED’,
                  position:P|_}
```

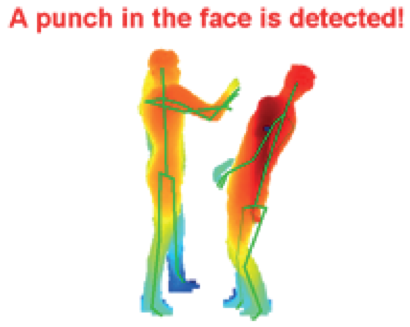

Fig. 2. *A punch in the face is detected by the logic program.*

```
          |_}
     |_}.
a_hand_of_a_person([_|R],Id,P):-
     a_hand_of_a_person(R,Id,P).
```

The following definition is used for the detection of the punch in the face. Execution of the is_near_a_head_of_a_colleague predicate succeeds, if given 3D coordinates of some person's hand are close enough to the coordinates of the head of another person. The predicate checks whether the person punches in the face of another person, but not in his own face. For brevity, the velocity of the movement of the hand and the duration of the contact are not taken into consideration, but in a real intelligent visual surveillance application these attributes are to be checked to provide acceptable specificity of the recognition procedure.

```
is_near_a_head_of_a_colleague
          ([S|_],Id1,P1):-
     S == {
          identifier:Id2,
               head:{
                   status:’TRACKED’,
                   position:P2|_}
          |_},
     Id1 <> Id2,
     is_near(P1,P2),!.
is_near_a_head_of_a_colleague
         ([S|_],Id1,P1):-
     S == {
          identifier:Id2,
          head:{
               status:’TRACKED’,
               position:P2|_}
          |_},
     Id1 <> Id2,
     is_near(P1,P2),!.
is_near_a_head_of_a_colleague([_|R],Id,P):-
    is_near_a_head_of_a_colleague(R,Id,P).
```

The is_near predicate is auxiliary. It checks whether the distance between the given co-ordinates is small enough. The predicate is defined using the built-in arithmetical predicates of Actor Prolog:

```
is_near({point:p(X1,Y1,Z1)|_},
        {point:p(X2,Y2,Z2)|_}):-
     Delta== ?sqrt(
          ?power(X1-X2,2) +
          ?power(Y1-Y2,2) +
          ?power(Z1-Z2,2)),
     Delta < 0.15,!.
```

For brevity, predicates implementing the data transferring from the outer world into the program are not considered here. One can find a detailed example of this kind in [50]. The analyse_skeletons predicate is to be called from time to time in the logic program to analyze images of skeletons recognized in 3D video frames transferred from the outside.

Note that the key point of this example is not recognition of the punch in the face (or recognition of a handshaking) that is a simple problem, but in that much more complex logical formulae can be automatically checked in the same way. For example, a logic program can check whether a person shakes hands with all people in the room or not, i.e. a program can make some conclusions on the social status of the person or social relations between all the people in the room. The same technique can be used to solve urgent problems of intelligent visual surveillance, such as recognizing armed attacks, demonstrating threatening poses, robberies, etc.

4. CONCLUSION

A method of the logic object-oriented programming for processing the data obtained from the equipment for 3D data acquisition in the intelligent video surveillance systems was developed. The Actor Prolog object-oriented logic language and translation of the logic programs of intelligent video surveillance to the Java language were used. The object-oriented means of the Actor Prolog logic language allow us to split a program into interacting parallel processes that implement various stages of image processing and scene analysis while translation to the Java language provides a high performance sufficient for analyzing real-time video, as well as reliability and stability of the work of the intelligent video surveillance software. One can download the software for free from the Web Site [52].

The application of methods of the logic object-oriented programming for 3D data processing gives exciting prospects in the area of intelligent video surveillance. Progress in this research area can be achieved by the development of efficient low-level 3D data processing algorithms, logic programming languages and systems for general purpose, and advanced logical methods of 3D data processing.

ACKNOWLEDGMENT

Authors are grateful to Ivan A. Kershner and Renata A. Tolmatcheva for the help in the preparation of 3D video samples and Angelos Barmpoutis for his J4K [59] library that was used for the data collection.

This research is supported by the Russian Foundation for Basic Research, grant No. 16-29-09626-ofi_m.

REFERENCES

1. Sebe IO, Hu J, You S, Neumann U. 3D video surveillance with augmented virtual environments *First ACM SIGMM international workshop on Video surveillance. ACM,* 2003:107-112.
2. Ott R, Gutiérrez M, Thalmann D, Vexo F. Advanced virtual reality technologies for surveillance and security applications. *International Conference on Virtual Reality Continuum and Its Applications. ACM*, 2006:163-170.
3. Barnum P, Sheikh Y, Datta A, Kanade T. Dynamic seethroughs: Synthesizing hidden views of moving objects. *Mixed and Augmented Reality (ISMAR 2009). IEEE*, 2009:111-114.
4. DeCamp P, Shaw G, Kubat R, Roy D. An immersive system for browsing and visualizing surveillance video. *International Conference on Multimedia. ACM,* 2010:371-380.
5. Gavrila DM. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 1999, 73(1):82-98.
6. Chen L, Wei H, Ferryman J. A survey of human motion analysis using depth imagery. *Pattern Recognition Letters,* 2013, 34(15):1995-2006.
7. Benedek C. 3D people surveillance on range data sequences of a rotating Lidar. *Pattern Recognition Letters,* 2014, 50:149-158.
8. Stettner R, Bailey H, Silverman S. Three dimensional Flash LADAR focal planes and time dependent imaging. *International Journal of High Speed Electronics and Systems,* 2008, 18(2):401-406.
9. Hansen DW, Hansen MS, Kirschmeyer M et al. Cluster tracking with time-of-flight cameras. *Computer Vision and Pattern Recognition Workshops* (*CVPRW'08*), *IEEE*, 2008:1-6.
10. Silvestre D. Video surveillance using a time-of-light camera. *Master's thesis, Informatics and Mathematical Modelling,* Technical University of Denmark, 2007.
11. Satta R, Pala F, Fumera G, Roli F. Real-time appearance-based person re-identification over multiple Kinect^{TM} cameras. *VISAPP*, 2013:407-410.
12. Sinha A, Chakravarty K, Bhowmick B. Person identification using skeleton information from Kinect. *The Sixth International Conference on Advances in Computer-Human Interactions* (*ACHI 2013*), 2013:101-108.
13. Krishnan P, Naveen S. RGB-D face recognition system verification using Kinect and FRAV3D databases. *Procedia Computer Science,* 2015, 46:1653-1660.
14. Diraco G, Leone A, Siciliano P. Human posture recognition with a time-of-flight 3D sensor for in-home applications. *Expert Systems with Applications,* 2013, 40(2):744-751.
15. Ibañez R, Soria Á, Teyseyre A, Campo M. Easy gesture recognition for Kinect. *Advances in Engineering Software*, 2014, 76:171-180.
16. Raheja J, Minhas M, Prashanth D et al. Robust gesture recognition using Kinect: A

comparison between DTW and HMM. *Optik International Journal for Light and Electron Optics*, 2015, 126(11):1098-1104.

17. Hsieh C-T, Wang H-C, Wu Y-K et al. A Kinect-based people-flow counting system. *Intelligent Signal Processing and Communication Systems* (*ISPACS 2012*). *IEEE*, 2012:146-150.
18. Aggarwal JK, Xia L. Human activity recognition from 3D data: A review. *Pattern Recognition Letters,* 2014, 48:70-80.
19. Lun R, Zhao W. A survey of applications and human motion recognition with Microsoft Kinect. *International Journal of Pattern Recognition and Artificial Intelligence,* 2015, 29(5).
20. Alazrai R, Mowafi Y, Lee C. Anatomical-plane-based representation for human–human interactions analysis. *Pattern Recognition*, 2015, 48(8):2346-2363.
21. Rougier C, Auvinet E, Rousseau J et al. Fall detection from depth map video sequences. *International Conference on Smart Homes and Health Telematics*. Springer, 2011:121-128.
22. Lee Y-S, Chung W-Y. Visual sensor based abnormal event detection with moving shadow removal in home healthcare applications. *Sensors*, 2012, 12(1):573-584.
23. Mastorakis G, Makris D. Fall detection system using Knect's infrared sensor. *Journal of Real-Time Image Processing,* 2014, 9(4):635-646.
24. Wang C, Liu H. Unusual events detection based on multi-dictionary sparse representation using Kinect. *Intern. Conf. on Image Processing, IEEE*, 2013:2968-2972.
25. Popa M, Koc AK, Rothkrantz LJ et al. Kinect sensing of shopping related actions. *International Joint Conference on Ambient Intelligence*, Springer, 2011:91-100.
26. Johanna M. Recognizing activities with the Kinect. A logic-based approach for the support room. *Master's thesis, Radboud University Nijmegen*, 2013.
27. Lau TB, Ong AC, Putra FA. Non-invasive monitoring of people with disabilities via motion detection. *International Journal of Signal Processing Systems,* 2014, 2(1):37-41.
28. Patwardhan A, Knapp G. Aggressive actions and anger detection from multiple modalities using Kinect. // arXiv preprint arXiv:1607.01076. – 2016.
29. Preis J, Kessel M, Werner M, Linnhoff-Popien C. Gait recognition with Kinect. *International Workshop on Kinect in PervasiveComputing.* Newcastle, UK, 2012.
30. Savage R, Clarke N, Li F. Multimodal biometric surveillance using a Kinect sensor. *Proceedings of the 12th annual security conference,* Las Vegas, USA, 2013.
31. Chaaraoui AA, Padilla-López JR, Flórez-Revuelta F. Abnormal gait detection with RGB-D devices using joint motion history features. *Automatic Face and Gesture Recognition* (*FG*), *IEEE,* 2015,7:1-6.
32. Leightley D, Yap MH, Hewitt BM, McPhee JS. Sensing behaviour using the Kinect: Identifying characteristic features of instability and poor performance during challenging balancing tasks. *Measuring Behavior*, 2016.
33. Borges PVK, Conci N, Cavallaro A. Video-based human behavior understanding: A survey. *IEEE Transactions on Circuits and Systems for Video Technology,* 2013, 23:1993-2008.
34. Haritaoglu I, Harwood D, Davis L. W^4: Who? When? Where? What? A real time system for detecting and tracking people. *FG 1998*, Nara, Japan, 1998:222-227.
35. Shet V, Harwood D, Davis L. VidMAP: Video monitoring of activity with Prolog. *AVSS-2005, IEEE*, 2005:224-229.
36. Shet V, Singh M, Bahlmann C et al. Predicate logic based image grammars for complex pattern recognition. I*nternational Journal of Computer Vision,* 2011, 93(2):141-161.
37. O'Hara S. VERSA–video event recognition for surveillance applications. *M.S. thesis. University of Nebraska at Omaha,* 2008.
38. Machot F, Kyamakya K, Dieber B, Rinner B. Real time complex event detection for resource-limited multimedia sensor networks. *AMMCSS-2011*, 2011:468-473.

39. Lao W, Han J, de With PHN. Flexible human behavior analysis framework for video surveillance applications. I*nternational Journal of Digital Multimedia Broadcasting*, 2010, Article ID 920121:1-9.
40. Artikis A, Sergot M, Paliouras G. A logic programming approach to activity recognition. *International Workshop on Events in Multimedia* (*EiMM 2010*), NY, USA. *ACM*, 2010:3-8.
41. Skarlatidis A, Artikis A, Filippou J, Paliouras G. A probabilistic logic programming event calculus. *Theory and Practice of Logic Programming*, 2014, 9:1-33.
42. Ferryman J, Hogg D, Sochman J et al. Robust abandoned object detection integrating wide area visual surveillance and social context. *Pattern Recognition Letters*, 2013, 34(7):789-798.
43. Katzouris N, Artikis A, Paliouras G. Event recognition for unobtrusive assisted living. *Hellenic Conference on Artificial Intelligence*, Springer, 2014:475-488.
44. Ruta M, Scioscia F, Summa MD et al. Semantic matchmaking for Kinect-based posture and gesture recognition. *International Journal of Semantic Computing*, 2014, 8(4):491-514.
45. Worch J-H, Bálint-Benczédi F, Beetz M. Perception for everyday human robot interaction. KI - Künstliche Intelligenz, 2016, 30(1):21-27.
46. Morozov AA, Vaish A, Polupanov AF et al. Development of concurrent object-oriented logic programming platform for the intelligent monitoring of anomalous human activities. *International Joint Conference on Biomedical Engineering Systems and Technologies*, Springer, 2015, 511:82-97.
47. Morozov AA, Polupanov AF. Intelligent visual surveillance logic programming: Implementation issues. *CICLOPS-WLPE 2014, ser. Aachener Informatik Berichte no. AIB-2014-09*, T. Ströder & T. Swift eds., RWTH Aachen University, 2014:31-45. http://aib.informatik.rwth-aachen.de/2014/2014-09.pdf.
48. Morozov AA, Polupanov AF. Development of the logic programming approach to the intelligent monitoring of anomalous human behaviour. *OGRW2014,* Koblenz, Univ. of Koblenz-Landau, 2015, 5:82-85. https://kola.opus.hbz-nrw.de/files/915/OGRW_2014_Proceedings.pdf.
49. Morozov AA, Sushkova OS, Polupanov AF. An approach to the intelligent monitoring of anomalous human behaviour based on the Actor Prolog object-oriented logic language. *RuleML 2015 DC and Challenge,* Berlin, CEUR, 2015. http://ceur-ws.org/Vol-1417/paper7.pdf.
50. Morozov AA. Development of a method for intelligent video monitoring of abnormal behavior of people based on parallel object-oriented logic programming. *Pattern Recognition and Image Analysis*, 2015, 25(3):481-492. http://www.cplire.ru/Lab144/pria481.pdf.
51. Morozov AA, Sushkova OS. Real-time analysis of video by means of the Actor Prolog language. *Computer Optics,* 2017; Special issue 3 (November): 97-105. DOI: 10.18287/2412-6179-2016-40-6-947-957.
52. Morozov AA, Sushkova OS. The intelligent visual surveillance logic programming Web Site [Electronic resource], 2017. Access mode: http://www.fullvision.ru (29.11.2017).
53. Morozov AA. Actor Prolog: an object-oriented language with the classical declarative semantics. IDL 1999, Eds. K. Sagonas, P. Tarau. Paris, France, 1999:39-53. http://www.cplire.ru/Lab144/paris.pdf.
54. Morozov AA. Logic object-oriented model of asynchronous concurrent computations. *Pattern Recognition and Image Analysis*, 2003, 13(4):640–649. http://www.cplire.ru/Lab144/pria640.pdf.
55. Morozov AA. Operational approach to the modified reasoning, based on the concept of repeated proving and logical actors. *CICLOPS 2007*, Eds. S. Abreu, V. S. Costa.

Porto, Portugal, 2007:1-15. http://www.cplire.ru/Lab144/ciclops07.pdf.

56. Morozov AA, Sushkova OS, Polupanov AF. A translator of Actor Prolog to Java. *RuleML 2015 DC and Challenge,* Eds. N. Bassiliades, P. Fodor, A. Giurca et al. Berlin, CEUR, 2015. https://www.csw.inf.fu-berlin.de/ruleml2015-ceur.
57. Morozov AA. A GitHub repository containing source codes of Actor Prolog built-in classes [Electronic resource]. 2017. Access mode: https://github.com/Morozov2012/actor-prolog-java-library.
58. Han J, Shao L, Xu D, Shotton J. Enhanced computer vision with Microsoft Kinect sensor: A review. *IEEE Transactions on Cybernetics,* 2013, 43(5):1318-1334.
59. Barmpoutis A. Tensor body: Real-time reconstruction of the human body and avatar synthesis from RGB-D. *IEEE Transactions on Cybernetics,* 2013, 43(5):1347-1356.
60. Han F, Reily B, Hoff W, Zhang H. Space-time representation of people based on 3D skeletal data: A review. ArXiv preprint arXiv:1601.01006v2 [cs.CV], 2016.
61. Presti LL, Cascia ML. 3D skeleton-based human action classification: A survey. *Pattern Recognition*, 2016, 53:130-147.