

[13] Real-Time analysis of video by means of the Actor Prolog language

A.A. Morozov ^{1,2}, O.S. Sushkova ¹

¹ Kotel'nikov Institute of Radio Engineering and Electronics of the RAS, Moscow, Russia

² Moscow State University of Psychology & Education, Moscow, Russia



Abstract

Object-oriented logical means for a real-time analysis of video were developed. The software includes a translator of the Actor Prolog object-oriented logic language to Java, an open source Java library that contains built-in classes of Actor Prolog, and a programming environment of the Actor Prolog language. An example of using the software for intelligent video monitoring of abnormal behavior in people is considered. Object-oriented means of the Actor Prolog language make it possible to divide a logic program into several parallel processes corresponding to various stages of the video processing. The translation of logic programs to Java provides reliability, portability, and openness of the intelligent video surveillance programs.

Keywords: *REAL-TIME VIDEO ANALYSIS, INTELLIGENT VISUAL SURVEILLANCE, PARALLEL OBJECT-ORIENTED LOGIC PROGRAMMING, ABNORMAL ACTIVITY DETECTION, ACTOR PROLOG, COMPLEX EVENT RECOGNITION, COMPUTER VISION, TECHNICAL VISION, TRANSLATION OF PROLOG TO JAVA.*

Citation: *MOROZOV AA, SUSHKOVA OS. REAL-TIME ANALYSIS OF VIDEO BY MEANS*

OF THE ACTOR PROLOG LANGUAGE. COMPUTER OPTICS 2016; 40(6): 947-957. DOI: 10.18287/2412-6179-2016-40-6-947-957.

Introduction

The objective of this work is to create a method and means for logic object-oriented programming (LOOP) of the next generation video surveillance systems. The idea behind intelligent video surveillance is automatic processing of the real-time data received from monitoring cameras in order to identify abnormal events including recognition of socially dangerous human behavior [1]. Typical tasks of the existing commercial intelligent video surveillance systems include: identification of abandoned items, presence of people in the controlled area, attendance estimates, fallen person, sudden change in the object movement behavior, attempts to tamper with the surveillance cameras etc. Further development of intelligent video surveillance equipment and means is linked to analyzing activities of single persons and groups [1, 2]. The following lines of research are related to human behavior [2]: gesture analysis, action analysis, behavior analysis, and analysis of interaction between humans and objects. The authors are first of all interested in behavior analysis and analysis of interaction between humans and objects. Other research field of the authors is analysis of test animals' behavior in biomedical science.

The key difference between human (animals) action analysis and behavior analysis is in that behav-

ior analysis covers the context of the actions taken. For example, the same action – employees quickly leaving the office – is normal in the end of the business day and during a lunch break, but during other hours of the day it may indicate some emergency and must be considered as abnormal behavior of the group of people. Using logic languages is a convenient and reasonable way to accumulate and apply the knowledge about the context of the observed events. Therefore, logic programming has been recently recognized as one of the most promising trends in behavior analysis [3 – 11].

The idea of logic programming for intelligent video surveillance systems is in using some logic language to describe and analyze video scenes and behavior scenarios of humans, animals, and technical units. In this case logical formulas describe object properties related to the problem to be solved and temporal and spatial relations between analyzed objects and events. Extensions to the existing logic languages and logic programming systems are generally used to implement this idea.

W⁴ [3] was one of the first intelligent video surveillance systems where logic programming was used to analyze temporal sequences that describe movements of people. Paper [4] describes another intelligent surveillance system, VidMAP, that combines C++ video image analysis algorithms and

Prolog logic programming system. VidMAP system ensures real-time video stream analysis, i.e. it processes video frames and makes logical inference at the same rate as the data is fed from video cameras. The system performs full cycle of video data processing including data input from one or several cameras, background removal, objects tracking, generation of logical formulas to describe the video scene, and logical inference. Paper [5] describes VERSA intelligent video surveillance system, which is based on SWI-Prolog logic programming system [12]. As opposed to VidMAP system VERSA does not provide full cycle of video images processing; logical inference is based on the pre-processed video frame descriptions. But VERSA includes the interval arithmetic based means to describe spatial and temporal relations between objects and events. Paper [6] describes the complex video and audio events recognition system based on Answer Set Programming [13]. Experiments with this system showed that this approach ensured high data processing speed sufficient to handle audio and video stream in real time.

One of the important research trends is developing fuzzy reasoning methods and means to analyze fuzzy and incomplete information about video scene and human, animal, and object behavior scenarios. Paper [7] describes the extension of predicate logic intended for fuzzy reasoning about observed objects and events. The predicate logic was augmented with bi-lattice formal description [14] that permits to handle fuzzy descriptions of observed images. Paper [8] describes another system based on the event calculus dialect [15] and the probabilistic logic programming system ProbLog [16]. The system input is fed with manually processed lists of so-called "short-term" events (walking, running, immobile condition, activity on the spot, and abrupt movements). Using probabilistic logical inference the system recognizes "long-term" human activities (a person left an item, meeting of people, joint movement of people, and a fight). One of the earlier works of the same research team [9] implemented the event calculus via standard logic programming means without probabilistic approach. Paper [8] discusses advantages and disadvantages of both approaches.

Paper [10] suggests a logical method of recognizing the acts of vandalism and hooliganism in the buses based on the video surveillance camera data. The audio and video data analysis is implemented via fuzzy logical inference. Paper [11] uses Prolog language to detect related groups of people in the crowd of passengers, which permits to identify ownership of

objects and thus reduce the number of abandoned objects false alarms.

Logic programming is widely used in research dedicated to analyzing video information stored in databases [17-26]. Semantics and video scene analysis descriptions use approximately the same methods as intelligent video surveillance but they emphasize big data analysis rather than real time data processing. First works in this area were not directly related to logic programming but they include easily recognizable ideas implemented in modern video logical analysis systems. For example, paper [18] describes NAOS system intended to recognize events based on geometric description of dynamic scenes. For scene description, this system uses first order logic formulas extended with temporal intervals, and for analysis, it uses the reference matching mechanism and search with rollback, which is characteristic for modern logic programming languages. Paper [19] describes SOO-PIN system designed to analyze dynamic scene semantics, including object movements and interactions. Dynamic scenes analysis implementation is based on the theory of frames by Marvin Minsky [27]. It is interesting to note that the system was developed on the basis of object-oriented parallel logic language Parlog++ [28], but the meaning of terms "object-oriented language" and "parallel logic language" as used by the authors of [19] significantly differs from the meaning applied in the present paper. This issue will be discussed below in more details. Paper [20] describes SQL-like language of spatial-temporal queries to video database. The language permits to describe spatial relations between objects using Prolog language rules. Paper [21] considers the intelligent video surveillance system with speech based human-machine interface that permits to ask questions about observed events including event time and location, gender of the person observed. Prolog language is used to detect events. Paper [22] suggests using Markov logic networks [29] for description and analysis of video data semantics. The idea behind this method of knowledge representation consists of using the first-order predicate logic formulas with weights that indicate their accuracy. A type of direct logical inference is used for logical analysis of video scenes. Later the proposed mathematical apparatus was successfully used in [23] to analyze complex interaction of a group of people using basketball game as an example. The Markov logic network apparatus is used in [24] for video recording of trauma surgeon team work. Paper [25] describes another method of fuzzy reasoning about a video scene. First order

logic language extended by subjective logic [30] is used to describe a video scene. Paper [26] describes the probabilistic event logic mathematical apparatus that combines the first order predicate logic with the set of time intervals and probabilistic estimates. Behavior of volleyball players is used as an example to discuss the developed method of video analysis. Despite intensive research in this field, logical methods and means of video images semantics analysis have been implemented only in experimental systems. One of the main reasons behind this is overall technical gap between logic programming and other information science disciplines. This gap is particularly vivid because there are much more people involved in creating and developing imperative languages and programming systems as compared to development of experimental and industrial logic programming systems. It means that successful evolution of this discipline would require not only development of new logical analysis methods and means, but also enhancement (improvement) of logic languages translators, programming systems, and application libraries. The authors consider this work as a step towards practical application of video images logical analysis methods, therefore logical analysis methods and means will be discussed together with their implementation.

The key feature of our approach to logic programming of intelligent video surveillance systems is using Actor Prolog object-oriented logic language [31-34]. The authors believe that support of object-oriented programming is essential to transition logical analysis methods from experimental research to practical application. Unfortunately, most of the works in the field of intelligent video surveillance logic programming are based on using classic (not object-oriented) Prolog (i.e. first-order predicate logic in the form of Horn clauses). There are a number of reasons behind this:

1. Developers of video analysis logical methods and means often ignore practical application issues.
2. The problem of integrating object-oriented and logical approaches to programming is a separate field of theoretic and experimental studies [35] and up until now there is no generally accepted approach to tackle this problem.
3. The most common open source logic programming systems are not object-oriented.

The challenges of integrating logic and object-oriented programming approaches lie in the requirement to develop mathematical semantics for the object-oriented means of logic language. There are three major approaches to logic and object-oriented programming integration [35]:

1. "Non-logical approach" where the problem of developing the mathematical semantics of the language is simply ignored. Syntactic constructions similar to object-oriented means of imperative languages are introduced into the language. The authors believe that this approach is dead-ended.
2. The so-called "clausal approach", where a set of Prolog language statements (Horn clauses) and logical variables is viewed as an "object".
3. "Process approach", where "objects" are recursive executable procedures. This approach was used in [19]. The authors believe that "process approach" is inconvenient in use and inefficient for implementation. For these reasons the logic programming means described in this paper are based on "clausal approach" to LOOP.

As compared to other programming languages, Actor Prolog [36, 37] has the following benefits:

1. All object-oriented language means have classical model-theoretic semantics. This means that there is a language subset for which standard theorems of logical inference soundness and completeness are fulfilled by resolution method. If certain conditions are satisfied it can be guaranteed that logic program will find all existing solutions of the problem and all found solutions will be mathematically correct.
2. Actor Prolog is a parallel logic language.
3. A translator of Actor Prolog to Java was developed [33, 34]. Logic language translation to Java is a reasonable choice for practical application of intelligent video surveillance methods, since it ensures reliability, portability, and openness of the created video processing systems.

This work describes the essentials of the method used to resolve intelligent video surveillance problem using recognition of abnormal human behavior as a case study.

1. Principals of Object-Oriented Logical Analysis of Video Images

Resolving the intelligent video surveillance problem can be split into low level recognition sub-problems (e.g. removal of video image background, segregation of persons and vehicles on the image, building the object movement trajectories, their speed estimates, etc.), and high level recognition sub-problems (e.g. recognition of running people, fights, abandoned items, etc.). Thereat logical means are convenient for high level recognition problems, and the input data for such means / algorithms shall be the low level recognition results obtained from the video image analysis methods implemented on lower level programming languages.

Parallel logic programming is very useful in terms of structuring the program code text and is absolutely essential for on-line analysis of video images because different analysis steps have different priorities. Some operations (e.g. background removal, selecting the blobs (blob – a set of inter-related pixels at the video image foreground), objects tracking, etc.) are critical because skipping these operations may cause failure to capture objects on the video image. Other operations (analysis of objects interaction graph, visualization of analysis results, etc.) may be suspended to save computing power. Unfortunately, most of logic languages, including Prolog, were originally created as consequential languages. Today parallel logic languages are not scarce but nevertheless the existing projects of logic programming for intelligent video surveillance are still based on sequential logic languages.

The so-called underdetermined sets of Actor Prolog [36] are used to describe objects (elements) of video scenes. For example, the underdetermined set $\{frame1: B, frame2: E, mean_velocity: V\}$ describes the section of the object movement trajectory within the time slot between frames B and E . Variable V is the mean object movement speed along this section.

Relations between video scene elements are described by Horn clauses, e.g. the rule

```
is_a_running_person(E):-
    E == { frame1: T1,
           frame2: T2,
           mean_velocity: V,
           mean_standardized_area: A,
           wr2_mean: M,
           wr2_skewness: S,
           wr2_cardinality: C | _ },
    is_a_fast_object(T1,T2,V),
    is_a_pedestrian(A,M,S,C).
```

will define the notion of a “running person” as a rapidly moving object, with the size and movement statistical metrics of this object matching a person (rather than a vehicle). Statistical metrics of the object movement will be discussed in section 3.

2. Example of Video Image Analysis

Let us review an example analysis of the video data from BEHAVE dataset [41] (Figure 1). Based on the logic programming method for intelligent video surveillance systems the program creates several parallel processes.



Fig.1. A situation similar to a street incident (attack of a group of hooligans) is in a view field of the intelligent video surveillance program. Rectangular boxes highlight moving blobs, thick lines designate connected graphs of blobs movements. The color of graph edge identifies the blobs speed

The first process reads the sequence of JPEG images, that emulates the real-time video input, and implements low-level analysis, i.e. background removal, blobs selection, objects tracking, identification of blobs interaction points. All these operations are performed via built-in Actor Prolog class 'ImageSubtractor', included into Java-library [39]. The second process analyzes the information pre-processed by the first process and displays the video surveillance results on the screen. Note that the second process must have lower priority as compared to the first one so that excessive utilization of computing resources by high level processing would not interrupt video capture in the first process.

The following scenario described by logical rules fits the purpose of recognizing abnormal human behavior in the example under review: "Somewhere in the camera field of vision two (or more) persons meet, after that a group splits and at least one per-

son runs." Such situation may be an evidence of a street incident, fight, or theft, so it should be highlighted as a possible case of abnormal behavior and a proper warning should be given to the video surveillance system operator.

The video data analysis method developed by the authors provides information about moving objects as Prolog terms (data structures) rather than Prolog facts (logical formulas). A logic program received the video image low-level analysis results as a data structure that describes the set of related graphs of blob movements. Each graph is represented as a list of underdetermined sets [36] that correspond to individual graph edges. Each edge is directed and provided with the following attributes: list of immediate predecessor edge numbers, list of direct successor edge numbers, coordinates and speed of a blob in different points of time, mean blob speed at this graph edge, etc.

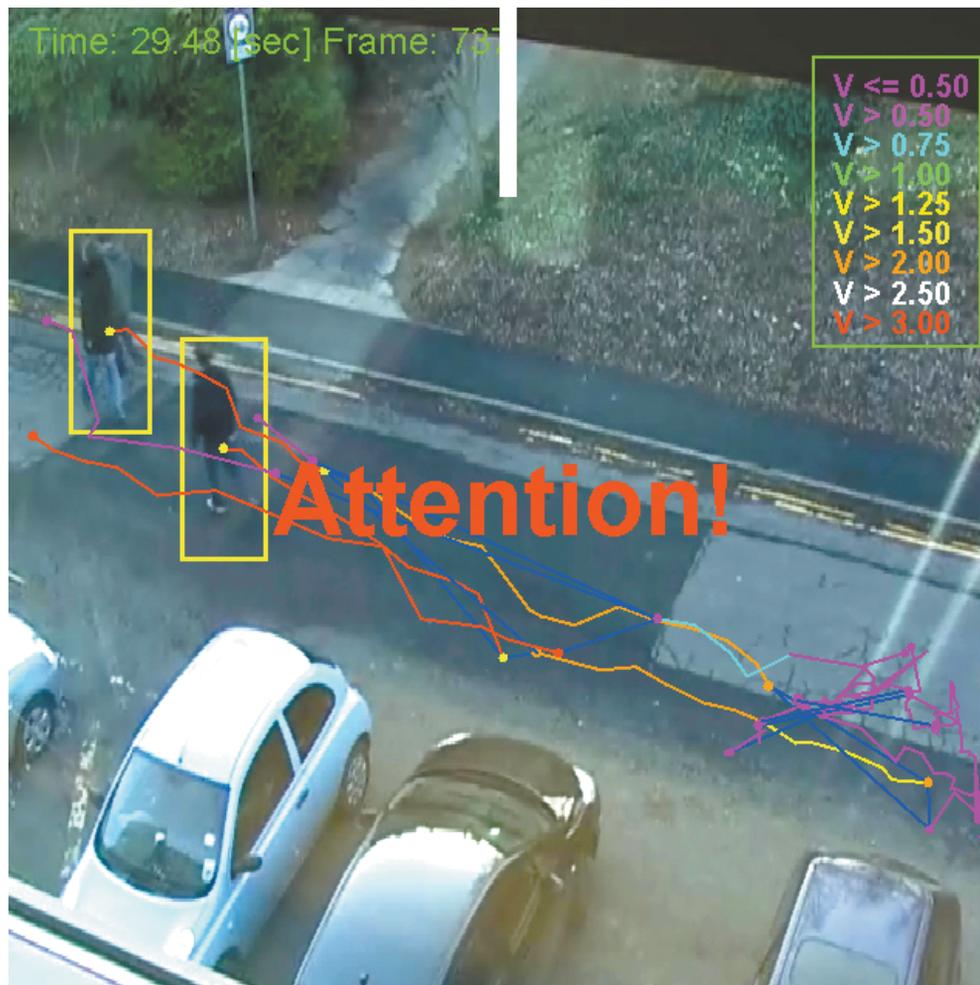


Fig.2. Intelligent video surveillance program has recognized a street incident. Rectangular boxes designate current position of the incident parties. Other designations are the same as on Figure 1

It is assumed that the graph describes the target situation if it contains at least one edge *A*, with successor edge *B* that corresponds to a running person, and predecessor edge *C* that corresponds to meeting of at least two persons. The graph edge corresponds to a running person if a mean speed and track length of the corresponding blob meets the pre-set logical condition (see [40] for the detailed example of a logical definition). If some graph contains the target street incident scenario, the logic program will display "Attention!" on the screen and will identify the current position of the incident parties (Figure 2).

The example under examination is a simple logic program that performs intelligent video surveillance of abnormal human behavior. Note that logic language has sufficient expression means to program all required steps of information processing: real time input of video information, low-level video processing, logical analysis of video information, and displaying results on the screen.

3. Experimental Verification of Logical Object-Oriented Analysis Method and Means

Experimental verification of intelligent video surveillance software is not a trivial task, since recognizing abnormal behavior includes a hierarchy of interrelated recognition sub-tasks, namely:

1. Accurate selection of moving objects.
2. Sufficiently accurate calculation of coordinates and estimate of object movement speed within a frame.
3. Classification of objects in the frame, as a minimum segregation of walking (running) persons and vehicles.
4. Search of the objects movement track graphs for sub-graphs that correspond to target abnormal behavior patterns.

Simple testing of recognition algorithms, namely counting correctly and incorrectly recognized test images with ROC-curves is not sufficient and can even be misleading as applied to software quality, because it does not provide understanding of the situation when minor changes of video surveillance conditions that downgrade quantitative characteristics of the low-level methods may affect the quality of higher level problem solutions. Therefore, full-scale testing of intelligent video surveillance software will require:

1. Creating special test video clips that reflect real life video surveillance conditions, including possible camera angles, changing illumination and background, line-up of the objects under surveillance, etc.
2. Comprehensive testing of individual steps of video data processing, with an understanding what param-

eters of one processing step would have a significant impact on implementing the next steps. For example, spikes in the estimates of actual objects velocity are critical to detect the so-called "abrupt movements" at the next step but can be ignored in the processing scheme described above in section 2.

3. Joint testing of all analysis steps including analysis system performance test at different load levels.

The authors are primarily interested in experimental verification and testing of new recognition methods and means developed within the scope of this project. The experimental verification results for one of the interim video analysis steps will be discussed below, namely segregation of moving objects into "persons" and "vehicles" classes.

Video dataset BEHAVE [41] provides good material for experiments with this sub-task. This data set contains staged fights, pursuits, and other kinds of abnormal behavior with several persons involved. Video clips are shot outdoors where real pedestrians, vehicles, and bicycles are moving.

The very first experiments with analyzing BEHAVE video examples demonstrated that moving vehicles cause numerous false alarms in abnormal behavior recognition logic programs, which are sensitive to object movement speed. Experiment results analysis showed that it is essential to ensure reliable segregation between fast moving persons and vehicles (including bicycles), but the available literature provide description only of inadequate classification methods based on recognizing the moving object shapes (e.g. two bicycle wheels) and (or) cyclical leg movements of a walking (running) person. Such methods are inadequate since moving objects often appear just for a short period or are partially covered by other objects.

Metrics that characterize changes of the moving objects shape were developed to resolve the recognition problem [32]. The metrics are based on the window modification of the determination coefficient R^2 , the mathematical meaning of this coefficient is a fraction of response variation Y , explained through the independent variable X in a certain regression equation. For example, if variable X means time, and variable Y – the length of the object contour, then R^2 value for a running person will be generally less than for a moving bicycle, because a running person is waving his hands and that results in constant changes in the length of his shape line without linear time dependence.

Mean value and skewness of R^2 values calculated in the non-overlapping windows were used as metrics to characterize moving objects.

Let us call a set of values below a window metrics wR^2 of some object movement track within the time interval from t_B to t_E

$$wR^2 = \{R_{i,w}^2\}, \quad (1)$$

where t – time ($t \in \{t_B + w/2, \dots, t_E - w/2\}$) and w – width of the window near the time moment t_i . Assume wR_i^2 – is the metrics R^2 value in the i -th window. Then the mean of the wR^2 value sample is

$$\text{mean}(wR^2) = (\sum_{i=1}^n wR_i^2)/n, \quad (2)$$

and the R^2 value sample skewness is

$$\text{skewness}(wR^2) = [\sqrt{n(n-1)/(n-2)}] \cdot s(wR^2), \quad (3)$$

$$s(wR^2) = \frac{\frac{1}{n} \sum_{i=1}^n (wR_i^2 - \overline{wR^2})^3}{\left(\sqrt{\frac{1}{n} \sum_{i=1}^n (wR_i^2 - \overline{wR^2})^2} \right)^3}, \quad (4)$$

where n – the number of windows at the examined section of the object movement trajectory.

Other factors being equal, vehicles as compared to pedestrians have greater values of metrics $\text{mean}(wR^2)$ (see Figure 3) and lesser values of metrics $\text{skewness}(wR^2)$. This permits to use these metrics in logical rules to segregate pedestrians from vehicles and bicycles, including fuzzy definitions.

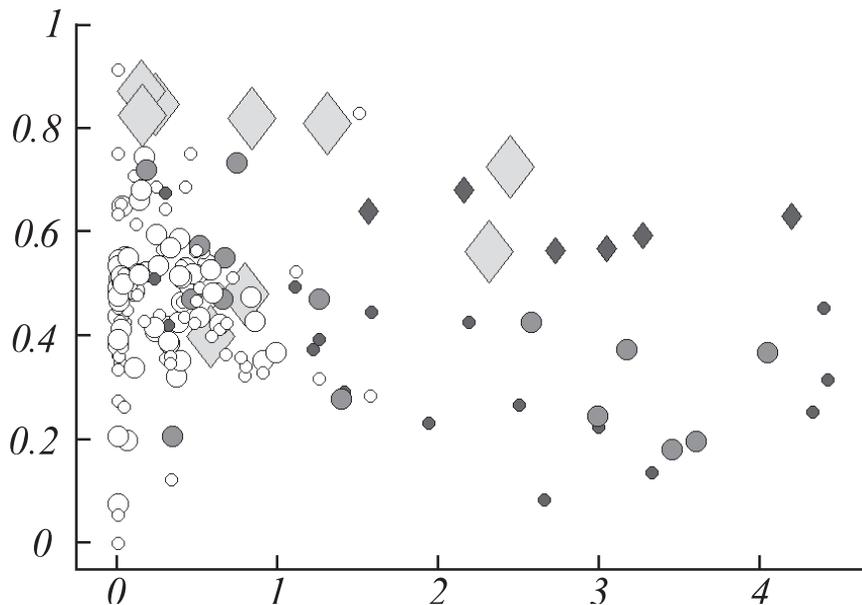


Fig.3. Value scattering of blobs movement metrics “mean by window R^2 ”. Metrics window length – 440 ms. Real time processing, Pentium 2.64 GHz. Coordinate x designates the object velocity, coordinate y designates the metrics value. Circles designate persons: small light circles show single walking persons, large light circles show groups of walking persons, small dark circles show single running persons, large dark circles show groups of running persons. Diamonds designate vehicles: small dark diamonds show cyclists, large light diamonds show motor vehicles

The key feature of the developed analysis method is implementation of fuzzy logical definitions via standard arithmetical predicates. They are used to predetermine threshold functions at various statistical metrics that characterize the size and shape changes of the objects. Model-theoretic semantics of logical formulas is preserved and regular SLD-resolution (depth-first search with backtracking) is used for logical inference, which ensures high information processing speed.

193 moving objects were selected from BEHAVE test set, the speed evaluation algorithm [31] selected 22 out of 193 blobs as fast objects. The next analysis stages used the developed moving objects size and

shape change metrics [32] to correctly classify all 22 blobs (all 15 running persons and 7 vehicles were recognized correctly).

4. Further Development of Object-Oriented Logical Analysis

Development of video image analysis method and means is currently in progress. Main activities in this field include:

1. Development of distributed logic programming means.
2. Improvement of low-level image analysis algorithms.

Means of distributed logic programming are created

for experimenting with multi-agent video information processing. The idea of multi-agent processing [42] originated from artificial intelligence science and implies that the video information collection and analysis system is broken down into a large number of independent programs (agents) that feature independence (the agent operates without direct involvement of a user or other agents), social networking (the agent interacts with other agents via certain preset mechanisms), reactivity (the agent responds to external events), and proactivity (the agent plans its activities to achieve certain goals).

Theoretically, using an agent-based approach to intelligent video surveillance systems development can ensure high flexibility, reliability, and scalability for such systems. For example, if individual agents perform different steps of low and high level video processing, a new type of analysis may be added to the system without modifying the existing agents and even without shutting down the system. To this end, a new agent is added to the system that can use the results of other agents and is able to communicate its own results to them.

The agents operating within the intelligent video surveillance system must perform specific functions

beyond conventional processing and symbolic information exchange operations characteristic for standard agent programming platforms. For these reasons in the intelligent video surveillance projects the researchers often develop their own agent platforms including Java-based ones, instead of trying to “squeeze” video processing into standard platforms.

Actor Prolog can be easily adapted to distributed intelligent video surveillance systems programming. This would not even require the language syntax expansion, possibility of remote method invocation for class instances from other programs will be sufficient [43].

We use Java remote method invocation (Java RMI) to implement this idea. The class instance received in the program from the outside can implement the remote predicate invocation, the same way as in the original instances of the program classes. Besides, the built-in classes of Actor Prolog responsible for image processing may use the class instances received from the outside for video data exchange.

The second field of work is related to video surveillance projects in biomedical experiments.

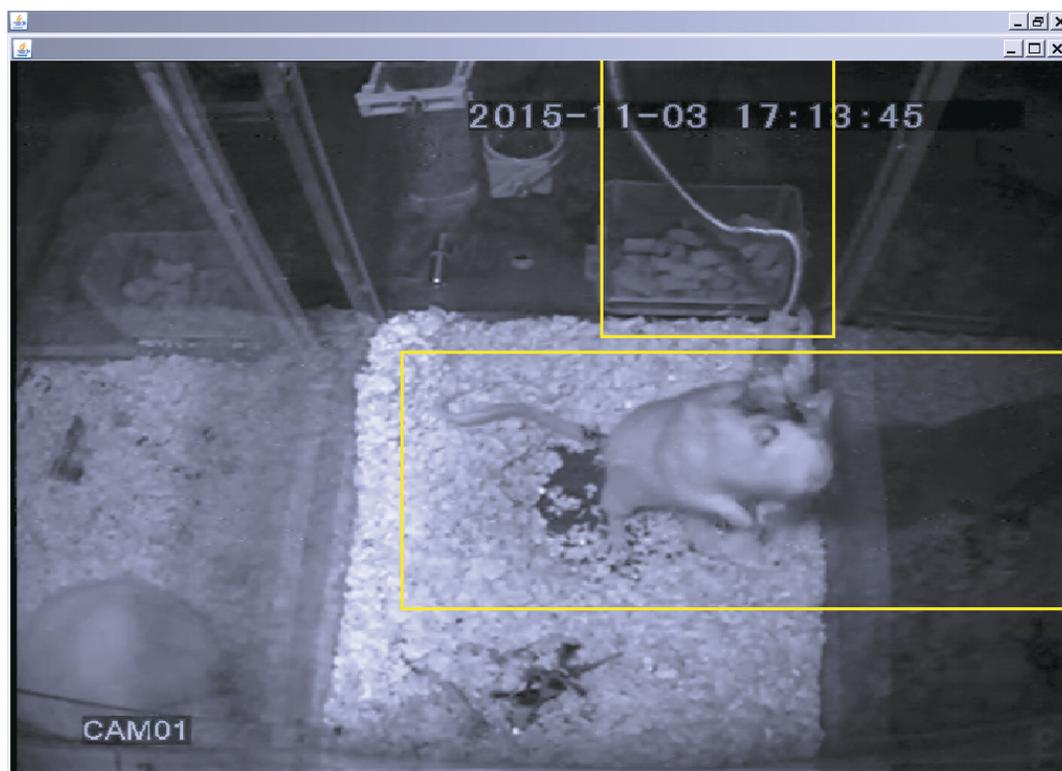


Figure 4. Video image of a test animal in a box. The rectangles indicate the blobs corresponding to an animal and the cable connecting the implanted electrodes with transmitters. Video is courtesy of Institute of Higher Nervous Activity and Neurophysiology of RAS

Figure 4 shows an example of video image of a test animal with implanted electroencephalographic (EEG) electrodes. The experiment objective was to study brain seizure activity, i.e. EEG of the test animal brain and animal behavior are recorded simultaneously. Comparing the EEG data and behavior is in particular required because fast movements of test animals may cause artefacts in EEG records, which are very similar to epileptic discharges. The video surveillance objective is to recognize animal fast movements and use this information for experiment results interpretation. The animal behavior is monitored round-the-clock including poor lighting environment. The animal fur color is only slightly different from the wood shredding on the cage floor; that is why precise segregation of the animal shape is a rather challenging task. Current developments include the background removal methods tolerant to poor lighting and lack of color information.

Conclusion

The paper discusses the object-oriented logic programming methods and means for intelligent video surveillance systems. The developed logic programming means include Actor Prolog object-oriented logic language translator to Java, open source library of built-in Actor-Prolog-Java classes [39] that includes low-level video analysis and computer graphics procedures, and Actor Prolog language programming environment. Object-oriented means of the logic language permit to split the program into interacting parallel processes that implement different stages of video processing and scene analysis; and translation to Java ensures reliability, portability, and openness of the created intelligent video surveillance programs. The developed software was published on-line [44] together with links to sample video data and articles dedicated to logic programming of video processing. The web-site [44] was created to support research in the field of logic programming of intelligent video surveillance system.

Acknowledgements

Authors are grateful to Abhishek Vaish, A.F. Polupanov, V.E. Antciperov, V.V. Deviatkov, A.N. Alfimtsev, V.S. Popov, I.I. Lychkov, A.O. Manolova, and I.G. Kormoltsev for co-operation.

The work was funded by the Russian Science Foundation (RSF), grant No. 16-11-10258.

References

1. Vishwakarma S, Agrawal A. A survey on activity recognition and behavior understanding in video surveillance. *Visual Computer* 2013; 29(10): 983-1009. DOI: 10.1007/s00371-012-0752-6.

2. Borges PVK, Conci N, Cavallaro A. Video-based human behavior understanding: A survey. *IEEE Transactions on Circuits and Systems for Video Technology* 2013; 23(11): 1993-2008. DOI: 10.1109/TCSVT.2013.2270402.

3. Haritaoglu I, Harwood D, Davis LS. W4: Who? When? Where? What? A real time system for detecting and tracking people. *Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998, Nara, Japan, 1998: 222-227. DOI: 10.1109/AFGR.1998.670952.

4. Shet V., Harwood D., Davis L. VidMAP: Video monitoring of activity with Prolog. *AVSS 2005*: 224-229. DOI: 10.1109/AVSS.2005.1577271.

5. O'Hara S. *VERSA – video event recognition for surveillance applications*. M.S. thesis. University of Nebraska at Omaha; 2008.

6. Machot F, Kyamakya K, Dieber B, Rinner B. Real time complex event detection for resource-limited multimedia sensor networks. *AVSS 2011*: 468-473. DOI: 10.1109/AVSS.2011.6027378.

7. Shet V, Singh M, Bahlmann C, Ramesh V, Neumann J, Davis L. Predicate logic based image grammars for complex pattern recognition. *International Journal of Computer Vision* 2011; 93(2): 141-161. DOI: 10.1007/s11263-010-0343-9.

8. Skarlatidis A., Artikis A., Filippou J., Paliouras G. A probabilistic logic programming event calculus. *Theory and Practice of Logic Programming* 2015; 15(2): 213-245. DOI: 10.1017/S1471068413000690.

9. Artikis A, Sergot M, Paliouras G. A logic programming approach to activity recognition. *EiMM '10 Proceedings of the 2nd ACM International Workshop on Events in Multimedia* 2010: 3-8. DOI: 10.1145/1877937.1877941.

10. Ma J, Liu W, Miller P, Yan W. Event composition with imperfect information for bus surveillance. *AVSS 2009*: 382-387. DOI: 10.1109/AVSS.2009.25.

11. Ferryman J, Hogg D, Sochman J, Behera A, Rodriguez-Serrano JA, Worgan S, Li L, Leung V, Evans M, Cornic P, Herbin S, Schlenger S, Dose M. Robust abandoned object detection integrating wide area visual surveillance and social context. *Pattern Recognition Letters* 2013; 34(7): 789-798. DOI: 10.1016/j.patrec.2013.01.018.

12. Wielemaker J, Schrijvers T, Triska M, Lager T. *SWI-Prolog. Theory and Practice of Logic Programming* 2012; 12(1-2): 67-96.

13. Baral C., Gelfond G., Son T.C., Pontelli E. Using answer set programming to model multi-agent scenarios involving agents' knowledge about other's knowledge. *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, Toronto, 2010; 259-266.

14. Ginsberg ML. Multivalued logics: A uniform approach to reasoning in artificial intelligence. *Computational intelligence* 1988; 4(3): 265-316. DOI: 10.1111/j.1467-8640.1988.tb00280.x.

15. Kowalski R, Sergot M. A logic-based calculus of events. *New Generation Computing* 1986; 4(1): 67-96. DOI: 10.1007/BF03037383.

- 16.** Kimmig A, Demoen B, Raedt LD, Costa VS, Rocha R. On the implementation of the probabilistic logic programming language ProbLog. *Theory and Practice of Logic Programming* 2011; 11(2-3): 235-262. DOI: 10.1017/S1471068410000566.
- 17.** Ballan L, Bertini M, Del Bimbo A, Seidenari L, Serra G. Event detection and recognition for semantic annotation of video. *Multi-media Tools and Applications* 2011; 51(1): 279-302. DOI: 10.1007/s11042-010-0643-7.
- 18.** Neumann B. Natural language description of time-varying scenes. *Semantic Structures: Advances in Natural Language Processing* 1984: 167-207.
- 19.** Dance S, Caelli T, Liu Z-Q. A concurrent, hierarchical approach to symbolic dynamic scene interpretation. *Pattern Recognition* 1996; 29(11): 1891-1903. DOI: 10.1016/0031-3203(96)00029-5.
- 20.** Dönderler ME, Ulusoy Ö, Güdükbay U. Rule-based spatio-temporal query processing for video databases. *The VLDB Journal* 2004; 13(1): 86-103. DOI: 10.1007/s00778-003-0114-0.
- 21.** QasemiZadeh B, Shen J, O'Neill I, Miller P, Hanna P, Stewart D, Wang H. A speech based approach to surveillance video retrieval. *AVSS '09 2009*: 336-339. DOI: 10.1109/AVSS.2009.54.
- 22.** Tran SD, Davis LS. Event modeling and recognition using Markov logic networks. *ECCV '08 2008; Pt II*: 610-623. DOI: 10.1007/978-3-540-88688-4_45.
- 23.** Morariu VI. Analyzing structured scenarios by tracking people and their limbs. Ph.D. Thesis. University of Maryland; 2010.
- 24.** Chakraborty I, Elgammal A. Video based activity recognition in a trauma center. 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG) 2013: 1-8.
- 25.** Han S., Koo B., Hutter A., Shet V.D., Stechele W. Subjective logic based hybrid approach to conditional evidence fusion for forensic visual surveillance. *AVSS 2010*: 337-344. DOI: 10.1109/AVSS.2010.19.
- 26.** Selman J, Amer M, Fern A, Todorovic S. PEL-CNF: Probabilistic event logic conjunctive normal form for video interpretation. *ICCV Workshops 2011*: 680-687. DOI: 10.1109/ICCVW.2011.6130308.
- 27.** Minsky M. A framework for representing knowledge. In book: Winston PH, Hom B, eds. *The psychology of computer vision*. New York: McGraw-Hill, 1975: 211-277.
- 28.** Davison A. From Parlog to Polka in two easy steps. In Book: *International symposium on programming language implementation and logic programming*. Berlin Heidelberg: Springer; 1991; 171-182. ISBN: 978-3-540-54444-9.
- 29.** Richardson M, Domingos P. Markov logic networks. *Machine Learning* 2006; 62(1): 107-136. DOI: 10.1007/s10994-006-5833-1.
- 30.** Josang A. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 2001; 9(3): 279-311. DOI: 10.1142/S0218488501000831.
- 31.** Morozov AA, Vaish A, Polupanov AF, Antciperov VE, Lychkov II, Alfmimtsev AN, Deviatkov VV. Development of concurrent object-oriented logic programming platform for the intelligent monitoring of anomalous human activities. In Book: Plantier G, Schultz T, Fred A, Gamboa H, eds. *Biomedical Engineering Systems and Technologies*. Springer International Publishing; 2015: 82-97. DOI: 10.1007/978-3-319-26129-4_6.
- 32.** Morozov AA, Polupanov AF. Development of the logic programming approach to the intelligent monitoring of anomalous human behaviour. In Book: Paulus D, Fuchs C, Droege D, eds. *Proceedings of 9th Open German-Russian Workshop on Pattern Recognition and Image Understanding (OGRW 2014)*. Koblenz: University of Koblenz-Landau; 2015: 82-85. Source: https://kola.opus.hbz-nrw.de/files/915/OGRW_2014_Proceedings.pdf.
- 33.** Morozov AA, Polupanov AF. Intelligent visual surveillance logic programming: Implementation issues. In Book: Ströder T, Swift T, eds. *Proceedings of the International Joint Workshop on Implementation of Constraint and Logic Programming Systems and Logic-based Methods in Programming Environments 2014 (July 17-18, 2014, Vienna, Austria)*. Aachen: RWTH Aachen University; 2014: 31-45. Source: <http://aib.informatik.rwth-aachen.de/2014/2014-09.pdf>.
- 34.** Morozov AA, Polupanov AF, Sushkova OS. A Translator of Actor Prolog to Java. In Book: Bassiliades N, Fodor P, Giurca A, Gottlob G, Kliegr T, Nalepa GJ, Palmirani M, Paschke A, Proctor M, Roman D, Sadri F, Stojanovic N, eds. *RuleML 2015 DC and Challenge*. Proceedings of the 9th International Rule Challenge and the 5th RuleML Doctoral Consortium (Berlin, Germany, August 2-5). Berlin: CEUR; 2015. Source: <https://www.csw.inf.fu-berlin.de/ruleml2015-ceur/>.
- 35.** Davison A. A survey of logic programming-based object-oriented languages. In Book: Wegner P, Yonezawa A, Agha G, eds. *Research Directions in Concurrent Object Oriented Programming*. Cambridge, MA: MIT Press; 1993: 42-106. ISBN: 0-262-01139-5.
- 36.** Morozov AA. Actor Prolog: An object-oriented language with the classical declarative semantics. In: Sagonas K, Tarau P, eds. *Proc IDL 1999*. Paris, France: 1999: 39-53. Source: <http://www.cplire.ru/Lab144/paris.pdf>.
- 37.** Morozov A.A. Logic object-oriented model of asynchronous concurrent computations. *Pattern Recognition and Image Analysis* 2003; 13(4): 640-649. Source: <http://www.cplire.ru/Lab144/pria640.pdf>.
- 38.** Chang C-L, Lee R. *Symbolic logic and mechanical theorem proving*. New York: Academic Press; 1973. ISBN: 978-0-08-091728-3.
- 39.** Actor-Prolog-Java-library: A GitHub repository containing source codes of Actor Prolog built-in classes. Source: <https://github.com/Morozov2012/actor-prolog-java-library/>.
- 40.** Morozov A.A. 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. DOI: 10.1134/S1054661815030153.
- 41.** Blunsden S.J., Fisher R.B. The BEHAVE video dataset: Ground truthed video for multi-person behavior classification. *Annals of the BMVA* 2010; 4: 1-12. DOI: 10.3390/s131114398.

- **42.** Gascueña JM., Fernández-Caballero A. On the use of agent technology in intelligent, multisensory and distributed surveillance. *The Knowledge Engineering Review* 2011; 26(2): 191-208. DOI: 10.1017/S0269888911000026.
- **43.** Morozov AA, Sushkova OS, Polupanov AF. Incorporation of distributed multiagent programming means in a strongly typed logic language [In Russian]. *Zhurnal Radioelektroniki [Journal of Radioelectronics]* 2016; 7. Source: <http://jre.cplire.ru/jre/jul16/9/text.pdf>.
- **44.** Development of logic programming approach to intelligent monitoring of anomalous human activities. Source: http://www.fullvision.ru/actor_prolog/show_eng.php