

# DEVELOPMENT OF LOGIC PROGRAMMING APPROACH TO INTELLIGENT MONITORING OF ANOMALOUS HUMAN ACTIVITIES<sup>1</sup>

Alexei A. Morozov

Kotel'nikov Institute of Radio Engineering and Electronics RAS  
Mokhovaya 11, Moscow, 125009, morozov@cplire.ru

A logic programming approach to the intelligent monitoring of anomalous human activities is considered. The main idea of this approach is using of a first order logic for describing abstract concepts of anomalous human activities (like tussle, leaving object, loitering, pickpocketing, personal theft, sudden attack, armed attack, immobile person, etc.). In the paper, we have described our first experiments in this area. We have used a Prolog-to-Java translator for efficient implementation of logical inference on video scenes. The problem of creation of special built-in classes of the Actor Prolog concurrent logic language for the low-level video processing is discussed.

## Introduction

Human activity recognition is a rapidly growing research area with important application domains including security and anti-terrorist issues [1-3]. In the past decade, logic programming was recognized as a promising approach for dynamic visual scenes analysis [4-7]. The idea of the logic programming approach is usage of logical rules for description and analysis of people activities. In this framework, knowledge about object coordinates / properties, scene geometry, and human body constraints is encoded in the form of rules in a logic programming language and applied to the output of low-level object / feature detectors. There are several research projects based on this idea. A. Artikis et al. [4] developed a system that recognizes so-called long-term activities (such as fighting and meeting) as temporal combinations of short-term activities (walking, running, inactive, etc.) using a logic programming implementation of the Event Calculus. Authors use the ProbLog state-of-the-art probabilistic logic programming framework to handle the uncertainty that occurs in human activity recognition. An obvious merit of this approach is a high level of abstraction in describing human activity. The underside of the approach is of little use

for the real time video processing. The mathematical semantics of probabilistic logical inference is also questionable. V. Shet et al. [5] have created an extension of predicate logic with the bilattice formalism that permits the handling of uncertainty in the reasoning. Earlier these authors have created the VidMAP visual surveillance system that combines real time computer vision algorithms with Prolog based logic programming. S. O'Hara [6] has created the VERSA general-purpose framework for defining and recognizing events in live or recorded surveillance video streams. VERSA provides more advanced spatial and temporal reasoning than VidMAP and is based on SWI-Prolog. F.A. Machot et al. [7] have proposed real time complex audio-video event detection based on Answer Set Programming approach. The authors proclaim that their solution is robust and can easily run on a chip.

A common approach to human activity recognition includes low-level and high-level recognition. An implementation of the logic programming approach requires consideration of several mathematical and engineering problems on each level of recognition:

1. Low-level processing stages usually include recognition of moving objects, pedestrians, faces, heads, guns, etc. The low-level procedures should be fast

---

<sup>1</sup> Supported by RFBR (13-07-92694-Ind\_a).

enough to be used for real time processing and the high-level logical means should utilize effectively and completely the output of the low-level procedures.

2. Adequate high-level logical means should be created / used to deal with temporal and spatial relationships in the video scene, as well as uncertainty in the results of low-level recognition procedures.
3. A proper hierarchy of the low-level and high-level video processing procedures should be constructed. The procedures of different levels are usually implemented on the base of different software means and programming languages.
4. Selection of logic programming system for implementation of the logical inference is of great importance, because it should provide high-level syntax means for implementation of required logical expressions as well as computation speed and efficiency in real time processing of big amount of data.

In our research, we address the problem of the high-level semantic analysis of people activities. We use the Actor Prolog concurrent logic programming language [8-14] for implementation of the logical inference on video scenes. The Actor Prolog is a kind of Prolog-to-Java translator that provides means for a high-level concurrent programming and a direct access to the low-level processing procedures written in Java. In the paper, we have described our first experiments in this area. The problem of creation of special built-in classes of the Actor Prolog logic language for the low-level video processing is discussed.

### Logical analysis of manually marked videos

On the first stage of the research, we have implemented several experiments on analysis of manually marked videos that is traditional approach in the area. The CAVIAR data sets (see the EC funded project IST 2001 37540, <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>) were used. The CAVIAR data sets are annotated using the XML-based Computer Vision Markup Language (CVML). The structure of CVML is simple enough, so we read it using the built-in class of the Actor Prolog for XML / HTML parsing. The CVML

annotations contain information about coordinates of separate persons and groups of persons in videos. So, our experiments have pursued the following goals:

1. To check if the Actor Prolog system is fast enough to process videos in real time even without performing low-level analysis.
2. To check if there is enough information about the positions of persons / groups for accurate estimation of the velocity and the acceleration of separate personages in the video scene.

The former issue is important because the accurate estimation of the velocity / acceleration opens a way for the recognition of so-called abrupt motions of objects [4]. This kind of motions is necessary for recognition of several long-term activities (such as fighting or sudden attacks), though recognition of abrupt motions is not usually provided by standard low-level analyzing procedures. The abrupt motions are not marked in the CAVIAR annotations as well.

An example of abrupt motion recognition is shown on the Fig. 1. A program written in Actor Prolog uses given coordinates of two persons to estimate the distance between them and the 2-nd derivative of the coordinates to detect abrupt motions of the persons.



Fig. 1. An example of CAVIAR video.

A logical rule describes an abnormal behavior (fighting) as a conjunction of two conditions:

1. Several persons have met sometime and somewhere.
2. After that they implement abrupt motions.

The text of the logic program is not given here for short. After recognition these two conditions, the logic programming system has

decided that there was a case of scuffle and has indicated the fighting persons by a red rectangle (see Fig. 2).



Fig. 2. The logic programming system has recognized that persons 6 and 7 formed a group and were fighting.

This example demonstrates a possibility of recognition semantics of video scenes using the logical inference on results of the low-level recognition of separate objects; however one can see the following bottle-neck of the approach. Manually defined coordinates of the objects were used for estimation of their acceleration and nobody can guarantee that automatic low-level procedures will provide the exact values of coordinates that are good enough for numerical differentiation. So, the discussion on the high-level recognition procedures is impossible without consideration of underlying low-level recognition methods.

The second issue of this example is whether it is useful to separate the recognition process into concurrent sub-processes implementing different stages of the high-level logical inference. Working intensity of different sub-processes is various. For example, the differentiation of coordinates requires more computational resources and another sub-process that implements recognition of people behavior could wait for the results of differentiation.

### Development of low-level procedures

A promising approach for implementation of the low-level recognition procedures is usage of the OpenCV computer vision library and we are planning to link Actor Prolog with the JavaCV library that is a Java interface to OpenCV. Nevertheless, Java has enough

standard means to solve simple image processing / recognition problems and we have started our experiments with pure Java.

We have created low-level Java procedures that implement several basic recognition tasks:

1. Background subtraction;
2. Discrimination of foreground blobs;
3. Tracking of the foreground blobs over time;
4. Detection of interactions between the blobs.

The first experiments have demonstrated clearly that the exact estimation of an object velocity was impossible without the accounting of the interactions of objects (see Fig. 3), because of edge effects of differentiation in the interaction points.



Fig. 3. A low-level procedure discriminates trajectories (red lines) of objects (violet rectangles) and moments of their interactions (green circle marks).

After implementation of this check we have got tracks that were accurate enough to determine if a person was walking or running. Nevertheless, the problem of abrupt motions detection is still a challenge. It is an evidence of that robust recognition of abrupt motions is impossible without the accurate low-level recognition of body parts (face, hands) and is a matter for the further investigation.

### Creation of a built-in class of Actor Prolog

At the next stage of research, we are planning to create a special built-in class of the Actor Prolog language that implements formerly described low-level recognition procedures.

The class will implement the following functions:

1. Acquisition of video frame sequences in the MPG and the JPEG formats.
2. Recognition of moving blobs and creation of Prolog data structures describing coordinates, size and color histograms of blobs in every moment of time.
3. Recognition of tracks of blob motions and creation of Prolog data structures describing coordinates, velocity and acceleration of blobs. The tracks are to be divided into separate segments, where segment ends there are points of interaction between the blobs.
4. Low-level means for analyzing attributes of tracks (length, mean velocity, etc.) and discrimination of tracks will be created (for instance, a problem of separation of slow moving and fast moving objects is to be considered). A fuzzy logic means are to be used for this purpose.

### Conclusion

Development of the logic programming approach to the problem of intelligent monitoring of people activities is a challenging problem and one can expect a rapid growth of research activity in this area. Development of high-level logical means requires severe consideration of recognition and performance capabilities of low-level procedures that provide prime statements for the logical inference. A prospective approach for implementation of the idea is using translation from a concurrent logic programming language to Java. Recent experiments with the Actor Prolog logic programming system have demonstrated that the concurrent logic language was suitable for this purpose and provided essential separation of the recognition process into concurrent sub-processes implementing different stages of the high-level analysis. On the next step of the research we are planning to create special built-in classes of the Actor Prolog language implementing the low-level analysis of video information. A prototype of an open source Java library will be created for experimenting with the logical description and analysis of people behavior, which can facilitate research

in the direction of intelligent monitoring of anomalous people activities.

### References

1. J.K. Aggarwal, M.S. Ryoo Human Activity Analysis: A Review // ACM Computing Surveys (CSUR). – 2011. – Vol. 43. – Issue 3 (April).
2. J. Junior, S. Musse, C. Jung Crowd analysis using computer vision techniques. A survey. // IEEE Signal Processing Magazine. – 2010. – September. – P. 66-77.
3. I.S. Kim, H.S. Choi, K.M. Yi, J.Y. Choi, S.G. Kong Intelligent Visual Surveillance – A Survey // International Journal of Control, Automation, and Systems. – 2010. – Vol. 8. – No. 5. – P. 926-939.
4. J. Filippou, A. Artikis, A. Skarlatidis, G. Paliouras A Probabilistic Logic Programming Event Calculus // Computing Research Repository. – 2012. – Vol. abs/1204.1851. – <http://arxiv.org/abs/1204.1851>.
5. V. Shet, M. Singh, C. Bahlmann, V. Ramesh, J. Neumann, L. Davis Predicate Logic Based Image Grammars for Complex Pattern Recognition // International Journal of Computer Vision. – 2011. – Vol. 93. – Issue 2 (June). – P. 141-161.
6. S. O'Hara, VERSA – Video event recognition for surveillance applications: M.S. Thesis. – University of Nebraska at Omaha, 2008. – 93 p.
7. F.A. Machot, K. Kyamakya, B. Dieber, B. Rinner Real Time Complex Event Detection for Resource-Limited Multimedia Sensor Networks // Workshop on Activity monitoring by multi-camera surveillance systems (AMMCSS). – 2011. – P. 468-473.
8. A.A. Morozov Actor Prolog: an Object-Oriented Language with the Classical Declarative Semantics // IDL'99. – Paris. – 1999.
9. A.A. Morozov On Semantic Link between Logic, Object-Oriented, Functional and Constraint Programming // MultiCPL'02. – Ithaca. – 2002. – P. 43-57.
10. A.A. Morozov Logic Object-Oriented Model of Asynchronous Concurrent Computations // Pattern Recognition and Image Analysis. – 2003. – Vol. 13. – No. 4. – P. 640-649.
11. A.A. Morozov Development and Application of Logical Actors Mathematical Apparatus for Logic Programming of Web Agents // ICLP 2003 Proceedings. – Springer-Verlag, 2003. – LNCS 2916. – P. 494-495.
12. A.A. Morozov Operational Approach to the Modified Reasoning, Based on the Concept of Repeated Proving and Logical Actors // CICLOPS. – Porto. – 2007. – P. 1-15.
13. A.A. Morozov Visual Logic Programming Method Based on Structural Analysis and Design Technique // ICLP 2007 Proceedings. – Springer-Verlag, 2007. – LNCS 4670. – P. 436-437.
14. A.A. Morozov Actor Prolog to Java translation // IIP-9, Montenegro, Budva. – Moscow: Torus Press, 2012. – P. 696-698. – In Russian.