An Approach to the Intelligent Monitoring of Anomalous Human Behaviour Based on the Actor Prolog Object-Oriented Logic Language

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

[1] Kotel'nikov Institute of Radio Engineering and Electronics of RAS, Mokhovaya 11, Moscow, Russia
[2] Moscow State University of Psychology & Education, Sretenka 29, Moscow, Russia
morozov@cplire.ru, sashap55@mail.ru, o.sushkova@mail.ru
http://www.fullvision.ru/

Abstract. A method for the monitoring of anomalous human behaviour that is based on the logical description of complex human behaviour patterns and special kinds of blob (a separated area of a foreground image) motion statistical metrics is developed. The concurrent object-oriented logic language is used for the analysis of graphs of tracks of moving blobs; the graphs are supplied by low-level analysis algorithms implemented in a special built-in class of Actor Prolog. The blob motion statistics is collected by the low-level analysis procedures that are of the need for the discrimination of running people, people riding bicycles, and cars in a video scene. The first-order logic language is used for implementing the fuzzy logical inference based on the blob motion statistics. A research software platform is developed that is based on the Actor Prolog logic language and a state-of-the-art Prolog-to-Java translator for experimenting with the intelligent visual surveillance.

1 Introduction

Human activity recognition is a rapidly growing research area with important application domains including security and anti-terrorist issues. Recently logic programming was recognized as a promising approach for dynamic visual scenes analysis (see surveys of logic-based recognition systems in [1,2]). The idea of the logic programming approach is in usage of logical rules for description and analysis of people activities. Knowledge about object co-ordinates and properties, scene geometry, and human body constraints is encoded in the form of certain rules in a logic programming language and is applied to the output of low-level object / feature detectors.

The distinctive feature of our approach to the visual surveillance logic programming is in the application of general-purpose concurrent object-oriented logic programming features, but not in the development of a new logical formalism. We use the Actor Prolog object-oriented logic language [3,4,2] for implementation of concurrent stages of video processing. A state-of-the-art Prolog-to-Java translator is used for efficient implementation of logical inference on video scenes.

Basic principles of video surveillance logic programming are described in Section 2. Development of the blob motion metrics for discrimination of running people, bicycles, and cars in the video scene is described in Section 3.

2 Basic principles of video surveillance logic programming

Let us consider an example of logical inference on video. The input of a logic program written in Actor Prolog is a standard BEHAVE [5] sample.

(a)

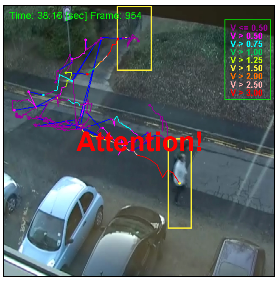

(b)


Fig. 1. (a) An example of BEHAVE [5] video with a case of a street offence: one group attacks another. (b) The logical inference has found a possible case of a street offence in the graph of blob trajectories. All probable participants of the conflict are marked by yellow rectangles. The tracks are designated by lines.

The video (see Fig. 1a) demonstrates a case of a street offence. Two groups of persons meet in the scope of the video camera, then one group attacks another one, they fight, then people run away.

We will solve the problem of anomalous human activity recognition using a logic program that describes a given scenario of complex people behaviour. The input data for the logic program will be supplied by low-level algorithms that trace objects in a video scene and estimate average speed in different segments of the trajectories [4,2]. This low-level processing is implemented in Java (not in Prolog) and includes extraction of foreground blobs, tracking of the blobs over time, detection of interactions between the blobs, creation of connected graphs of linked tracks of the blobs, and estimation of average speed of the blobs in separate segments of the tracks (see Fig. 2a). The input data include a special set of blob motion metrics to discriminate running pedestrians, bicycles, and cars during the logical inference.

The logic program checks the graph of tracks and looks for the following pattern of interaction among several persons: “If two or more persons meet somewhere in the scene and one of them runs after the end of the meeting, the program should consider this scenario as a kind of a running away and a probable case of a sudden attack or a theft.” Thus, the program has to alarm if this kind of sub-graph is detected in the total connected graph of tracks. In this

(a) (b)

Fig. 2. (a) The low-level processing of the video. Blobs are designated by cyan rectangles; multicoloured lines denote tracks of the blobs. The program estimates the velocity of the blobs and designates it by different colours. Straight blue lines designate possible links between the blobs. (b) The wR^2 metrics is computed on the basis of temporal changes of the length of the contour of the blob.

case, the program marks all persons in the inspected graph by yellow rectangles and outputs a warning in the middle of the screen (see Fig. 1b).

3 Development of the blob motion metrics

In contrast to the previously developed motion analysis methods of pedestrian detection that consider the periodicity and cyclic motion in the way humans movement [6], we have developed a set of blob motion statistical metrics based on the windowed coefficient of determination of the temporal changes of the length of the contour of the blob [7].

The coefficient of determination R^2 [8] indicates the proportionate amount of variation in the given response variable Y explained by the independent variable X in a linear regression model. In the general case, vehicles will be characterised by bigger values of the R^2 metrics than running persons, because the contour of the running person changes permanently in the course of his motion when he waves his arms and moves his legs (see Fig. 2b).

We use a windowed modification of the R^2 metrics, that is, the trajectory of a moving blob is characterised by a set of instantaneous values of the R^2 metrics computed in each point of the trajectory. Specifically, two statistical metrics that characterise the motion of the blob, namely, the mean of the wR^2 distribution and the bias-corrected skewness of the wR^2 distribution are used.

The properties of the $mean(wR^2)$ and the $skewness(wR^2)$ metrics can be illustrated by the example of the BEHAVE data set [5]. For that, we have computed the tracks of moving blobs in this data set by the blob extraction methods implemented in Actor Prolog [4,2]. The "speed-$mean(wR^2)$" and the "speed-$skewness(wR^2)$" diagrams show clearly that these metrics allow discriminating fast moving persons and vehicles (Fig. 3). We have implemented the metrics in the Vision standard package of the Actor Prolog language and use them for experimenting with the intelligent visual surveillance.

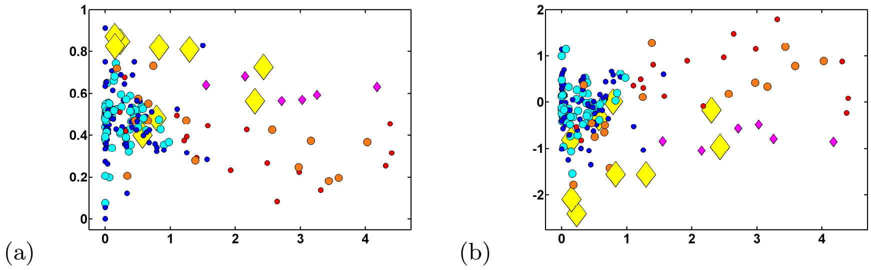


Fig. 3. (a) The values of the $mean(wR^2)$ metrics of the blobs. The x co-ordinate is the speed of the object; the y co-ordinate is the $mean(wR^2)$ metrics value. (b) The values of the $skewness(wR^2)$ metrics of the blobs. The x co-ordinate is the speed of the object; the y co-ordinate is the $skewness(wR^2)$ metrics value. In both figures, pedestrians are designated by circles: small blue circles denote single walking persons; big cyan circles denote groups of walking persons; small red circles denote single running persons; big orange circles denote groups of running persons. Vehicles are designated by diamonds: small magenta diamonds denote bicycles and big yellow diamonds denote cars.

We have incorporated elements of fuzzy logical inference based on the blob motion statistics into the logic language by means of standard arithmetic means of the language. For example, the $is\_a\_fast\_object$ predicate for recognition of fast objects may take into account simultaneously two characteristics of blobs; namely, the average velocity of a blob and the length of a track. Combination of these two characteristics is performed by very simple fuzzy metrics described in terms of arithmetic functions. From the standpoint of the declarative semantics of the language, the procedure is a standard formula of the first-order predicate logic. Please see a detailed example of a logic program in [7].

We use a compilation from the Actor Prolog language to Java [2] to ensure necessary performance of logical inference as well as reliability, portability, and openness of the software, including the possibility of access to open source code libraries. At present, we can demonstrate that the Actor Prolog system is fast enough for real-time analyzing clips of the standard data set [5].

4 Conclusions

We have developed a research software platform based on the Actor Prolog concurrent object-oriented logic language and a state-of-the-art Prolog-to-Java translator for studying the intelligent visual surveillance. The platform includes the Actor Prolog logic programming system and an open source Java library of Actor Prolog built-in classes [9]. It is intended to facilitate the study of the intelligent monitoring of anomalous people activities, the logical description and analysis of people behaviour (see Web Site [10]).

Special blob motion statistical metrics based on the windowed coefficient of determination wR^2 were developed. The blob motion statistics are collected by the low-level analysis procedures that are implemented in a specialised built-in class of the Actor Prolog system. Our studies have demonstrated that this approach is a prospective one for description and analysis of complex people behaviour, and is useful in recognition of anomalous people activities.

Acknowledgements

Authors are grateful to Abhishek Vaish, Vyacheslav E. Antciperov, Vladimir V. Deviatkov, Aleksandr N. Alfimtsev, Vladislav S. Popov, and Igor I. Lychkov for co-operation.

We acknowledge a partial financial support from the Russian Foundation for Basic Research, grant No 13-07-92694.

References

1. Skarlatidis, A., Artikis, A., Filippou, J., Paliouras, G.: A probabilistic logic programming event calculus. Theory and Practice of Logic Programming (2014) 1–33
2. Morozov, A.A., Polupanov, A.F.: Intelligent visual surveillance logic programming: Implementation issues. In Ströder, T., Swift, T., eds.: CICLOPS-WLPE 2014. Number AIB-2014-09 in Aachener Informatik Berichte, RWTH Aachen University (2014) 31–45
3. Morozov, A.A.: Actor Prolog: an object-oriented language with the classical declarative semantics. In Sagonas, K., Tarau, P., eds.: IDL 1999, Paris, France (1999) 39–53
4. Morozov, A.A., Vaish, A., Polupanov, A.F., Antciperov, V.E., Lychkov, I.I., Alfimtsev, A.N., Deviatkov, V.V.: Development of concurrent object-oriented logic programming system to intelligent monitoring of anomalous human activities. In Jr., A.C., Plantier, G., Schultz, T., Fred, A., Gamboa, H., eds.: BIODEVICES 2014, SCITEPRESS (2014) 53–62
5. Fisher, R.: BEHAVE: Computer-assisted prescreening of video streams for unusual activities. The EPSRC project GR/S98146. [Online] Available from: http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/ (2013)
6. Borges, P.V.K.: Pedestrian detection based on blob motion statistics. IEEE Trans. Circuits Syst. Video Techn. (2013) 224–235
7. Morozov, A.A., Polupanov, A.F.: Development of the logic programming approach to the intelligent monitoring of anomalous human behaviour. In Paulus, D., Fuchs, C., Droege, D., eds.: OGRW2014, Koblenz, University of Koblenz-Landau (2015) 82–85
8. Draper, N., Smith, H.: Applied Regression Analysis. Wiley-Interscience (1998)
9. Morozov, A.A.: A GitHub repository containing source codes of Actor Prolog built-in classes. [Online] Available from: https://github.com/Morozov2012/actor-prolog-java-library (2014)
10. Morozov, A.A., Sushkova, O.S.: The intelligent visual surveillance logic programming Web Site. [Online] Available from: http://www.fullvision.ru/actor_prolog/ (2014)