

# АНАЛИЗ ВИДЕОИЗОБРАЖЕНИЙ В РЕАЛЬНОМ ВРЕМЕНИ СРЕДСТВАМИ ЯЗЫКА АКТОРНЫЙ ПРОЛОГ

А.А. Морозов<sup>1,2</sup>, О.С. Сушкова<sup>1</sup>

<sup>1</sup> Институт радиотехники и электроники им. В.А. Котельникова РАН, Москва, Россия,

<sup>2</sup> Московский городской психолого-педагогический университет, Москва, Россия

Разработано программное обеспечение (ПО) для анализа видеоизображений в реальном времени средствами логического программирования. В состав ПО входят: транслятор объектно-ориентированного логического языка Актерный Пролог в Джаву, свободно распространяемая библиотека встроенных классов Актерного Пролога на Джаве, включающая процедуры низкоуровневого анализа видеоизображений и компьютерной графики, а также среда программирования на языке Актерный Пролог. Использование ПО рассмотрено на примере решения задачи интеллектуального видеомониторинга аномального поведения людей. Объектно-ориентированные средства логического языка Актерный Пролог позволяют разбивать программу анализа видео на взаимодействующие параллельные процессы, реализующие различные этапы обработки видеоизображений и анализа сцен, а трансляция в язык Джавы обеспечивает надёжность, переносимость и открытость создаваемых программ интеллектуального видеомониторинга.

**Ключевые слова:** анализ видеоизображений в реальном времени, интеллектуальное видеонаблюдение, параллельное объектно-ориентированное логическое программирование, выявление аномального поведения, Актерный Пролог, распознавание сложных событий, машинное зрение, техническое зрение, трансляция Пролога в Джаву.

## Введение

Актуальность разработки методов и средств анализа видеоизображений в реальном времени в настоящее время значительно возросла в связи с широким распространением технологий интеллектуального видеонаблюдения. Идея интеллектуального видеонаблюдения заключается в автоматической обработке поступающего в реальном времени видеопотока с камер видеонаблюдения с целью распознавания аномальных событий и, в том числе, общественно-опасного поведения людей [1]. Одним из перспективных направлений в области распознавания аномального поведения в последние годы было признано логическое программирование интеллектуального видеонаблюдения [2-8] (см. также обзоры в работах [2,9]).

Идея логического программирования интеллектуального видеонаблюдения заключается в том, что для описания и последующего анализа сценариев аномального поведения людей и технических объектов используется некоторый логический язык; при этом с помощью логических правил описываются временные и пространственные связи между анализируемыми объектами и элементарными событиями, а также свойства объектов, имеющие отношение к решаемой задаче распознавания аномального поведения.

В статье рассматривается пример использования ПО, созданного для анализа видеоизображений в реальном времени на основе метода логического программирования интеллектуального видеонаблюдения [7,8]. В состав ПО входят: транслятор [9,10] объектно-ориентированного логического языка Актерный Пролог [11,12] в Джаву, свободно распространяемая библиотека встроенных классов [13] Актерного Пролога на Джаве, включающая процедуры низкоуровневого анализа видеоизображений и компьютерной графики, а также среда программирования на языке Актерный Пролог. Объектно-

ориентированные средства логического языка позволяют разбивать программу на взаимодействующие параллельные процессы, реализующие различные этапы обработки видеоизображений и анализа сцен [14], а трансляция в язык Джава обеспечивает надёжность, переносимость и открытость создаваемых программ интеллектуального видеомониторинга.

## **1. Принципы анализа видеоизображений**

При решении задачи интеллектуального видеонаблюдения можно выделить подзадачи распознавания низкого уровня (например, вычитание фона видеоизображения, выделение на видеоизображении людей, автомобилей, построение траекторий перемещения объектов, распознавание позы и отдельных частей тела человека, оценка скорости его перемещения, обнаружение резких движений и т.п.) и подзадачи распознавания высокого уровня (например, распознавание драки, вооружённого нападения, кражи, оставленных без присмотра предметов и др.). При этом логические средства удобно применять для решения высокоуровневых задач распознавания, а входными данными для них должны служить результаты распознавания низкого уровня, полученные с применением методов анализа видеоизображений, реализованных на языках программирования более низкого уровня.

Применение параллельного логического программирования является очень удобным с точки зрения структурирования текста программы и становится совершенно необходимым в условиях анализа видеоизображений, поступающих в реальном времени, потому что различные этапы анализа имеют разный приоритет. Некоторые операции (например, вычитание фона, выделение блоков, трассировка объектов и др.) являются критически важными, потому что их пропуск может привести к потере видеозахвата объектов. Другие операции (анализ графа взаимодействия объектов, визуализация результатов анализа и пр.) при этом могут быть приостановлены для экономии вычислительных ресурсов. К сожалению, большинство логических языков, включая первый логический язык Пролог, изначально создавались как последовательные языки. В настоящее время параллельные логические языки давно уже не являются экзотикой, но, тем не менее, существующие проекты логического программирования интеллектуального видеонаблюдения до сих пор были основаны на последовательных логических языках.

## **2. Пример анализа видеоизображений**

Рассмотрим пример анализа видеоданных из набора ВЕНАВЕ [15] (рис. 1). В соответствии с методом логического программирования интеллектуального видеонаблюдения [7,8], программа анализа создаёт несколько параллельных процессов.

Первый процесс читает последовательность JPEG-изображений, имитирующую ввод видеоизображения в реальном времени, и реализует анализ низкого уровня, а именно, вычитание фона, выделение блоков, трассировку объектов, определение точек взаимодействия блоков. Все эти операции реализуются с помощью встроенного класса Акторного Пролога 'ImageSubtractor', входящего в состав Джава-библиотеки [13]. Второй процесс анализирует информацию, подготовленную первым процессом, и выводит на экран ре-

зультаты видеонаблюдения. Заметим, что приоритет второго процесса при этом должен быть ниже, чем приоритет первого процесса, чтобы чрезмерное потребление вычислительных ресурсов на высокоуровневую обработку не могло привести к потере видеозахвата первым процессом.



**Рис. 1.** В поле зрения программы интеллектуального видеонаблюдения попала ситуация, похожая на уличный инцидент (нападение группы хулиганов). Прямоугольниками обозначены движущиеся блобы, жирными линиями обозначены связанные графы перемещений блобов. Скорость движения блобов обозначена с помощью цвета рёбер графа.

Для распознавания аномального поведения в рассматриваемом примере подходит следующий сценарий поведения, описанный в виде логических правил: «Где-то в поле видимости камеры встречаются два человека (или несколько человек), после чего группа людей разделяется, и, по крайней мере, один из людей убегает». Такая ситуация может свидетельствовать об уличном инциденте, драке или краже, поэтому её стоит отметить как вероятный случай аномального поведения и выдать соответствующее предупреждение оператору системы видеонаблюдения.

Логическая программа получает результаты низкоуровневого анализа видеоизображения в виде структуры данных, описывающей набор связанных графов перемещений блобов. При этом каждый граф представлен в виде списка недоопределённых множеств [11], соответствующих отдельным рёбрам графа. Каждое ребро является направленным и снабжено следующими атрибутами: список номеров рёбер - непосредственных предшественников рассматриваемого ребра, список номеров рёбер - прямых продолжателей рассматриваемого ребра, координаты и скорость блоба в различные моменты времени, а также средняя скорость блоба на данном ребре графа.



**Рис. 2.** Программа интеллектуального видеонаблюдения распознала уличный инцидент. Жёлтыми прямоугольниками обозначено текущее положение участников происшествия. Остальные обозначения такие же, как на рис. 1.

Считается, что граф описывает искомую ситуацию, если в нём содержится хотя бы одно ребро А, имеющее в качестве последователя ребро В, соответствующее бегущему человеку, и в качестве предшественника – ребро С, соответствующее встрече, по крайней мере, двух людей. Ребро графа соответствует бегущему человеку, если средняя скорость и длина трека блага, соответствующего этому ребру, отвечают заданному логическому условию (подробный пример логического определения см. в [14]). Если некоторый граф содержит искомый сценарий уличного инцидента, логическая программа выводит на экран надпись «Attention!» и указывает текущее положение участников происшествия (рис. 2).

Рассмотренный пример представляет собой простую логическую программу, осуществляющую интеллектуальный видеомониторинг аномального поведения людей. Заметим, что выразительных средств логического языка достаточно, чтобы запрограммировать все необходимые этапы обработки информации: ввод видеoinформации в реальном времени, низкоуровневая обработка видеоизображения, логический анализ видеoinформации и вывод результатов на экран.

### 3. Экспериментальная проверка ПО

Экспериментальная проверка ПО интеллектуального видеонаблюдения является нетривиальной задачей, потому что распознавание аномального поведения включает некоторую иерархию взаимосвязанных подзадач распознавания, а именно:

1. Аккуратное выделение движущихся объектов.
2. Достаточно точное вычисление координат и оценка скорости движения объектов в кадре.
3. Классификация объектов, присутствующих в кадре, по крайней мере, разделение идущих (бегущих) людей и транспортных средств.
4. Поиск подграфов, соответствующих искомым схемам аномального поведения, в графе траекторий движения объектов.

Простое тестирование алгоритмов распознавания, а именно, подсчёт правильно и неправильно распознанных тестовых образцов с построением ROC-кривых, является совершенно недостаточным и, более того, может ввести в заблуждение относительно качества работы ПО, потому что не даёт представления о том, как небольшие изменения условий видеонаблюдения, ухудшающие количественные характеристики работы методов низкого уровня, могут повлиять на качество решения задач более высокого уровня. Поэтому для полноценного тестирования ПО интеллектуального видеонаблюдения необходимо:

1. Создание специальных тестовых видеоклипов, отражающих реальные условия видеонаблюдения, включая возможные ракурсы съёмки, возможные изменения освещения и фона, состав наблюдаемых объектов и пр.
2. Всестороннее тестирование отдельных этапов обработки видеоданных с пониманием того, какие параметры одного этапа обработки существенно влияют на реализацию последующих этапов. Например, наличие выбросов в оценке мгновенной скорости объектов критично при детектировании на последующем

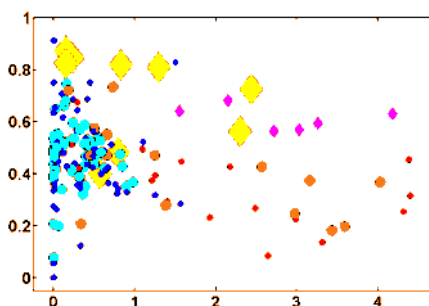
этапе так называемых «резких движений», но может быть проигнорировано в схеме обработки, описанной выше, в разделе 2.

3. Совместное тестирование всех этапов анализа, включая проверку производительности системы анализа при различных уровнях нагрузки.

Для авторов представляет интерес, прежде всего, экспериментальная проверка и тестирование работы новых методов и средств распознавания, разработанных в рамках данного проекта. Ниже будут рассмотрены результаты экспериментальной проверки одного из промежуточных этапов анализа видеоизображений, а именно, различение классов движущихся объектов «люди» и «транспортные средства».

Для экспериментов с указанной подзадачей распознавания хорошо подходит набор видеоданных BEHAVE [15]. В этом наборе содержатся инсценировки драк, преследования и других видов аномального поведения с участием нескольких людей. При этом съёмка осуществляется на улице, по которой время от времени проходят настоящие пешеходы, движутся автомобили и велосипедисты.

Первые же эксперименты с анализом видео-примеров BEHAVE продемонстрировали, что передвижение транспортных средств приводит к многочисленным ложным срабатываниям логических программ распознавания аномального поведения, учитывающих скорость движения объектов. Анализ результатов экспериментов показал, что совершенно необходимо надёжное различение быстро передвигающихся людей и транспортных средств (включая велосипедистов), однако описанные в литературе методы классификации, основанные на распознавании формы движущихся объектов (например, двух колёс велосипеда) и (или) циклического движения ног идущего (бегущего) пешехода, являются недостаточными, так как движущиеся объекты часто видны лишь в течение коротких промежутков времени или перекрываются другими объектами.



**Рис. 3.** Диаграмма рассеяния значений метрики движения блобов «среднее по оконному  $R^2$ ». Длина окна метрики 440 мс. Реальное время, Pentium 2.67 ГГц. Координата  $x$  обозначает скорость объекта, координата  $y$  обозначает величину метрики. Люди обозначены кружками: маленькие синие кружки обозначают одиночных пешеходов, большие голубые кружки обозначают группы пешеходов, маленькие красные кружки обозначают одиночных бегущих людей, большие оранжевые кружки обозначают группы бегущих людей. Транспортные средства обозначены ромбами: маленькие фиолетовые ромбы обозначают велосипедистов, большие жёлтые ромбы обозначают автомобили.

Для решения задачи распознавания были разработаны метрики, характеризующие изменения формы движущихся объектов [8]. Метрики основаны на оконной модификации множественного коэффициента детерминации  $R^2$ , математический смысл которого – доля вариации отклика  $Y$ , объяснённая независимой переменной  $X$  в некотором регрессионном уравнении. Например, если переменная  $X$  обозначает время, а переменная  $Y$  –

длину контура движущегося объекта, величина  $R_2$ , в общем случае, для бегущего человека будет меньше, чем для движущегося велосипеда, потому что бегущий человек машет руками, и это приводит к постоянному изменению длины его контура (не зависящему линейно от времени).

В качестве метрик, характеризующих движущиеся объекты, мы использовали среднее и асимметрию выборки значений  $R_2$ , вычисленных в неперекрывающихся окнах (см. рис. 3). В наборе тестов VEHAVE было выбрано 193 движущихся объектов, алгоритм оценки скорости объектов [7] выделил 22 блока из 193 как быстрые объекты. На последующих этапах анализа все эти 22 блока были правильно классифицированы как люди или транспортные средства с помощью разработанных метрик изменения формы движущихся объектов (распознаны 15 бегущих людей и 7 транспортных средств).

### **Заключение**

Созданное ПО опубликовано на сайте [16] вместе со ссылками на примеры видеоданных и статьями о логическом программировании видеообработки. Целью создания сайта [16] является поддержка исследований в области логического программирования интеллектуального видеонаблюдения.

### **Благодарности**

Авторы выражают благодарность Abhishek Vaish, ИТ Allahabad, Александру Ф. Полупанову и Вячеславу Е. Анциперову, ИРЭ им. В.А. Котельникова РАН, а также Владимиру В. Девяткову, Александру Н. Алфимцеву, Владиславу С. Попову и Игорю И. Лычкову, МГТУ им. Н.Э. Баумана, Москва.

Исследование выполнено за счёт гранта Российского научного фонда (проект № 16-11-10258).

### **Литература**

1. Vishwakarma S., Agrawal A. A survey on activity recognition and behavior understanding in video surveillance // *Visual Computer*. – 2013. – Vol. 29, No. 10. – P. 983-1009.
2. Skarlatidis A., Artikis A., Filippou J., Paliouras G. A probabilistic logic programming event calculus // *Theory and Practice of Logic Programming*. – 2014. – No. 9. – P. 1-33.
3. 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. – Vol. 93, No. 2. – P. 141-161.
4. Shet V., Harwood D., Davis L. VidMAP: Video monitoring of activity with Prolog // *AVSS 2005*. – IEEE, 2005. – P. 224-229.
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 // *AMMCSS 2011*. – 2011. – P. 468-473.
7. Morozov A.A., Vaish A., Polupanov A.F. Antciperov V.E., Lychkov I.I., Alifimtsev A.N., Deviatkov V.V. Development of concurrent object-oriented logic programming platform for the intelligent monitoring of anomalous human activities // *BIOSTEC 2014 / Ed. by G. Plantier, T. Schultz, A. Fred, H. Gamboa*. – CCIS 511. – Springer International Publishing, 2015. – P. 82-97.
8. Morozov A.A., Polupanov A.F. Development of the logic programming approach to the intelligent monitoring of anomalous human behaviour // *OGRW 2014 / Ed. by D. Paulus, C. Fuchs, D. Droege*. – Ko-

- blenz: University of Koblenz-Landau, 2015. – No. 5. – P. 82-85. – [https://kola.opus.hbz-nrw.de/files/915/OGRW\\_2014\\_Proceedings.pdf](https://kola.opus.hbz-nrw.de/files/915/OGRW_2014_Proceedings.pdf) .
9. Morozov A.A., Polupanov A.F. Intelligent visual surveillance logic programming: Implementation issues // CICLOPS-WLPE 2014 / Ed. by T. Ströder, T. Swift. – Aachener Informatik Berichte no. AIB-2014-09. – RWTH Aachen University, 2014. – P. 31-45. – <http://aib.informatik.rwth-aachen.de/2014/2014-09.pdf> .
  10. Morozov A.A., Polupanov A.F., Sushkova O.S. A Translator of Actor Prolog to Java // RuleML 2015 / Ed. by N. Bassiliades, P. Fodor, A. Giurca, G. Gottlob, T. Kliegr, G.J. Nalepa, M. Palmirani, A. Paschke, M. Proctor, D. Roman, F. Sadri, N. Stojanovic. – Berlin: CEUR, 2015. – <https://www.csw.inf.fu-berlin.de/ruleml2015-ceur/> .
  11. Morozov A.A. Actor Prolog: an object-oriented language with the classical declarative semantics // IDL 1999 / Ed. by K. Sagonas, P. Tarau. – Paris, France: 1999. – P. 39-53. – <http://www.cplire.ru/Lab144/paris.pdf> .
  12. Morozov A.A. Logic object-oriented model of asynchronous concurrent computations // Pattern Recognition and Image Analysis. – 2003. – Vol. 13, No. 4. – P. 640-649. – <http://www.cplire.ru/Lab144/pria640.pdf> .
  13. Morozov A.A. A GitHub repository containing source codes of Actor Prolog built-in classes. – 2016. – <https://github.com/Morozov2012/actor-prolog-java-library/> .
  14. 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. – Vol. 25, No. 3. – P. 481-492.
  15. Fisher R. BEHAVE: Computer-assisted prescreening of video streams for unusual activities. The EPSRC project GR/S98146. – 2013. – <http://groups.inf.ed.ac.uk/vision/BEHAVEDATA/INTERACTIONS/>.
  16. Morozov A.A., Sushkova O.S. The intelligent visual surveillance logic programming Web Site. – 2016. – <http://www.fullvision.ru> .