



ISSN 2071-2898 (Print)  
ISSN 2071-2901 (Online)

**Соколов С.С., Богуславский А. А.**

Автоматизация подготовки  
требований к программному  
обеспечению систем  
компьютерного зрения  
реального времени

**Рекомендуемая форма библиографической ссылки:** Соколов С.С., Богуславский А. А. Автоматизация подготовки требований к программному обеспечению систем компьютерного зрения реального времени // Препринты ИПМ им. М.В.Келдыша. 2016. № 141. 16 с. doi:[10.20948/prepr-2016-141](https://doi.org/10.20948/prepr-2016-141)  
URL: <http://library.keldysh.ru/preprint.asp?id=2016-141>

**Ордена Ленина  
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ  
имени М.В.Келдыша  
Российской академии наук**

**С.С. Соколов, А.А. Богуславский**

**Автоматизация подготовки требований  
к программному обеспечению  
систем компьютерного зрения  
реального времени**

**Москва — 2016**

УДК 004.932.2

*Соколов С.С., Богуславский А.А.*

### **Автоматизация подготовки требований к программному обеспечению систем компьютерного зрения реального времени**

Управление требованиями – необходимый этап при разработке ПО. В системах компьютерного зрения реального времени наиболее существенные требования связаны с обработкой видеопоследовательностей. Требования такого типа особенно трудоемки для формулировки и формализации. В связи с распространением задач систем технического и компьютерного зрения становится актуальной задача управления требованиями к ПО СТЗ по обработке и классификации видеопоследовательностей. Основными этапами в процессе разработки ПО СТЗ реального времени являются: подбор представительных видеопоследовательностей для решаемой задачи, классификация видеопоследовательностей (на основе видов фоноцелевой обстановки, обычно формулируемых на естественном языке), формализация требований для каждого класса, разработка алгоритма для каждого класса. Целью проводимых исследований является сокращение временных затрат на разработку ПО СТЗ РВ от постановки задачи на естественном языке до реализации алгоритмов обработки зрительных данных, решающих целевую задачу в масштабе реального времени. В работе описывается первый этап решения рассматриваемой задачи, на котором был проведен анализ имеющихся средств для автоматизированной обработки текстовых описаний сцен. Рассмотрены возможности их приложения к формализации требований к ПО СТЗ РВ.

**Ключевые слова:** система технического зрения реального времени, проектирование программного обеспечения, управление требованиями, обработка зрительных данных в режиме реального времени

*Sokolov Sergei Sergeevich, Boguslavsky Andrey Alexandrovich*

### **Automation of requirements preparation for the software of real-time vision systems.**

Requirements management is a necessary stage in the software development. In real-time computer vision systems, the most essential requirements are connected with video sequences processing. Requirements of this kind are especially time-consuming for the formulation and formalization. Due to the distribution of computer vision systems there is actual a problem of requirements management for processing and classification of video sequences. The main stages in the real-time vision system software development are: selection of representative video sequences for a solved applied task, classification of video sequences (on the basis of the types of a background and target images which are usually formulated in a natural language), formalization of requirements for each class of video sequences, development of algorithm for each class. The purpose of described researches is reduction of time costs for vision systems software development. In this work the first stage of the solution of a considered task is described.

**Key words:** real-time vision system, software design, software requirements, real-time image processing

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проекты №№ 16-08-01282, 16-37-00490 и Российского научного фонда, проект № 16-19-10705.

## **Введение**

В процессе проектирования и реализации систем технического и компьютерного зрения реального времени одним из наиболее важных этапов является начальный этап формирования требований к программной системе. Характерной особенностью рассматриваемых программных систем является слабая формализация входных данных, представленных в виде последовательностей изображений.

На сегодняшний день задача автоматизации обработки описаний видеопоследовательностей в системах компьютерного зрения реального времени полностью ещё не автоматизирована вследствие недостаточной унификации описаний, в связи с проблемами с извлечением знаний из текстов на естественном языке и спецификой обработки зрительных данных в реальном времени.

Применительно к управлению требованиями к ПО СТЗ РВ можно сформулировать задачу наглядного представления требований в унифицированном и структурированном виде. В работе рассматриваются способы графового представления требований по результатам обработки текстовой информации на естественном языке. Для визуализации графов доступно большое количество программных продуктов, форматов файлов и способов задания графов. В работе рассматриваются основные средства и методы описания и визуализации графов.

## **Автоматическая обработка текстовых данных**

Методы анализа текстовых данных постоянно совершенствуются и улучшаются. Поисковые системы в интернете, вопросно-ответные системы и прочие CASE-средства стремятся извлекать максимум данных из текста на естественном языке. Суперкомпьютер и соответствующее ему ПО IBM Watson, вероятно, наиболее известно в плане анализа текстовых описаний [1]. Методы, применяемые в UIMA, на которой основан Watson, и сама архитектура слишком сложны и избыточны для задач разработки систем компьютерного зрения. Настраивать универсальную систему синтаксического, лексического и прочих анализов – трудоёмкая и длительная подзадача, после решения которой основная задача обработки видео в режиме реального времени на основе данных, извлечённых из текстовых описаний, так и останется открытой. Поэтому для решения задачи автоматизации анализа текстовых описаний видеопоследовательностей, помимо разработки программы обработки текстовых описаний видеопоследовательностей, была произведена разработка программы визуализации полученных на предыдущем этапе данных, произведено составление типовых спецификаций, которым должны соответствовать описания видеопоследовательностей. Подобные спецификации

в данный момент используются применительно к требованиям в различных международных стандартах [2-4].

Известные алгоритмы обработки текстовых данных и программы на их основе предполагают достаточно высокую степень стандартизации и формализации данных. К сожалению, описания видеоданных, как правило, даются в свободном формате, удобном и понятном конечному пользователю, но не программисту и аналитику. В качестве компромисса предлагается разбивать видеопоследовательности на сцены по некоторым характерным признакам (таким как расстояние, характерная точка, чётко различимые границы, статичные объекты в кадре) и строить текстовое описание сцен. Компромисс подразумевает, что описание должно быть написано на естественном языке и должно быть понятно конечному пользователю и, в то же время, быть достаточно структурированным и формализованным, чтобы имелась возможность автоматической обработки этого описания для построения семантических структур данных, описывающих сцены, и дальнейшего их использования. Один из вариантов дальнейшего использования полученных семантических структур данных: построение графа, визуализирующего объекты на сцене и их отношения.

### **Пример требований к программному обеспечению СТЗ реального времени**

В качестве примера требований к ПО СТЗ РВ рассмотрим задачу слежения за процессом сближения и стыковки КК и МКС к стыковочному узлу СМ.

Для контроля параметров относительного движения КК и МКС в процессе сближения и стыковки по данным изображений, поступающим с ТВ-камеры, которая установлена на КК «Прогресс», на МКС на торце служебного модуля в окрестности стыковочного узла установлена стыковочная мишень. Параметры изображения этой мишени, полученные в ПО СТЗ, используются расчетным модулем для определения параметров положения корабля относительно станции (дальность и угловые отклонения) на расстоянии менее 15 м. Мишень представляет собой окружность с внешним радиусом 0.4 м (рис. 1). На окружности нанесены угловые деления – метки в виде квадратов и прямоугольников. Плоскость мишени совпадает с плоскостью торца служебного модуля станции. В центре мишени перпендикулярно ее плоскости расположен стержень длиной 0.6 м. На конце стержня установлен прямой крест. Плоскость креста параллельна плоскости мишени, размах каждой его перекладины – 0.2 м.

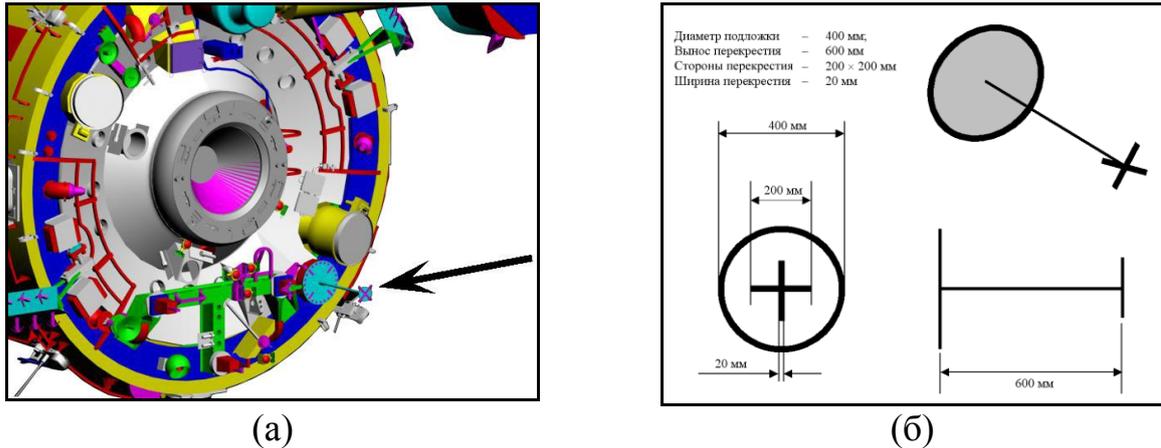


Рис.1. (а) Расположение стыковочной мишени на МКС (на изображении трехмерной модели служебного модуля мишень отмечена стрелкой).

(б) Устройство и размеры стыковочной мишени (угловые деления на схеме не показаны).

При идеальной стыковке центр креста на изображении должен совпадать с центром мишени, а концы перекладин креста – находиться против выделенных меток в форме прямоугольников. В реальном процессе (рис. 5) эта конфигурация нарушена. По размеру мишени, по горизонтальному и вертикальному смещениям ее центра относительно центра креста на видеоизображении можно найти положение фокуса камеры относительно станции. По смещениям центра мишени относительно центра кадра и по углу поворота креста относительно границ кадра можно найти три угла, задающие ориентацию связанной с камерой системы координат относительно системы координат станции. Таким образом, вблизи станции видеоизображение мишени и креста дает исчерпывающую информацию о положении корабля. Имея последовательность кадров с измерениями указанных смещений и углов, оказывается возможным восстановление движения корабля относительно станции.

На расстоянии более 15 м по ТВ-изображениям достаточно надежно определяется только расстояние между кораблем и станцией. Такое определение выполняется по размеру окружности поперечного сечения станции (т.е. по изображению торца служебного модуля), при дополнительном предположении, что торец станции расположен примерно перпендикулярно оптической оси камеры. С учетом этого предположения также принимается, что соответствующие оси систем координат мишени и камеры параллельны. Анализ большого числа стыковок показывает, что такое предположение допустимо.

На расстояниях менее 15 м СТЗ должна выполнять обнаружение стыковочной мишени и ее составных частей: перекрестия и меток. Эти данные передаются расчетному модулю для построения модели движения КК и определения параметров относительного движения КК и МКС на основе этой модели. На расстояниях между КК и МКС более 15 м стыковочная мишень на

ТВ-изображении видна неустойчиво – угловые деления на изображении отсутствуют или имеют площадь порядка нескольких пикселей, толщина сторон перекрестия на изображении также становится сравнима с размерами пиксела. Для обеспечения определения дальности и скорости КК относительно МКС на расстояниях более 15 м СТЗ выполняет обнаружение на изображениях поперечного сечения станции. Таким образом, в рассматриваемой СТЗ можно выделить два образа, требующих обнаружения: образ поперечного сечения станции и образ стыковочной мишени.

На основе доступных тестовых видеопоследовательностей на этапе проектирования и опытной эксплуатации СТЗ были сформированы наборы изображений, соответствующих расстоянию между КК и МКС с шагом 5 м и 1 м. В результате анализа этих наборов изображений в интервале прослеживаемых расстояний (от 100 до 0 м) были выделены четыре класса изображений, на которых можно указать однотипные наборы признаков для обнаружения образа станции и ее частей (рис. 1). Эти классы изображений соответствуют четырем различным фрагментам прослеживаемого интервала. Ниже перечислены характерные особенности этих изображений, которые учитываются при проектировании алгоритмов обработки в СТЗ.

Дальность от 100 м до 45 м (рис. 2): на изображениях заметны солнечные батареи и поперечное сечение станции. Внешний контур стыковочного узла виден неустойчиво, в зависимости от условий освещения при выполнении стыковки. В центральной части образа стыковочного узла присутствует темная область. Стыковочная мишень практически не видна.

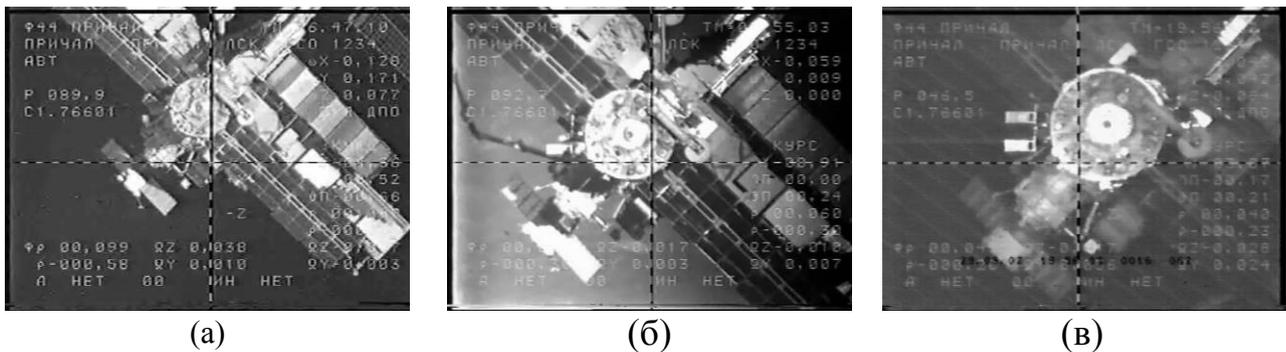
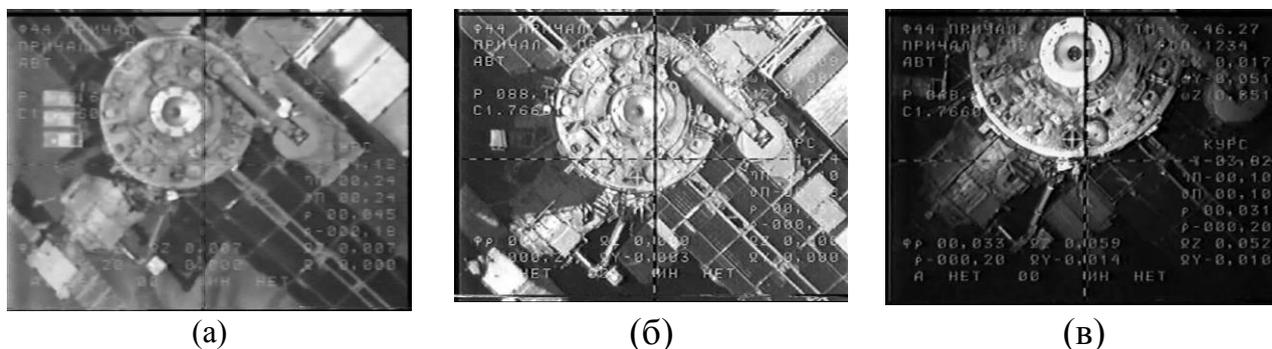


Рис. 2. Изображения МКС, полученные в процессе различных сеансов стыковки, типичные при дальности корабля в интервале от 100 м до 45 м.  
(а) Дальность 100 м; (б) 65 м; (в) 55 м.

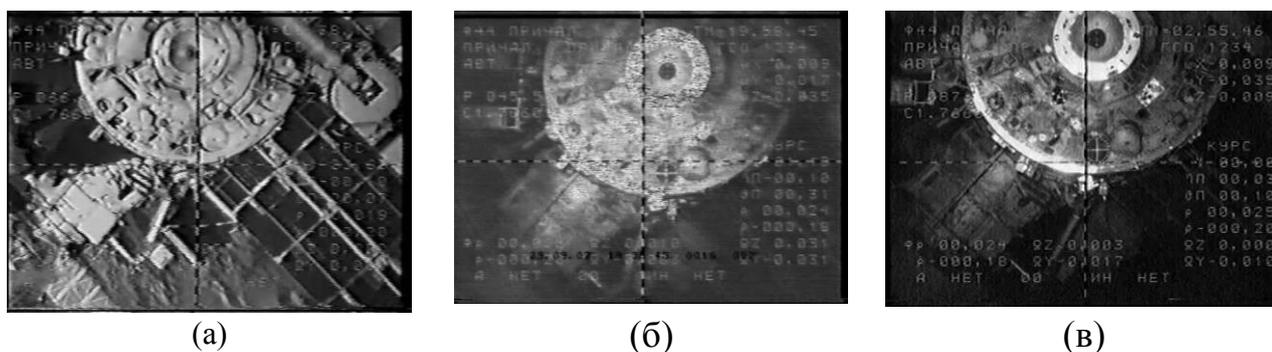
В интервале расстояний от 45 м до 25 м условия наблюдения позволяют выделить стыковочный узел как совокупность связанных областей и контуров, образующих круговой образ на изображении (рис. 3). На образе поперечного сечения станции становятся заметны контрастные контуры частей станции, установленных на ее торце. Образы этих деталей, а также неуправляемое освещение приводит к формированию на образе поперечного сечения большого количества областей различной яркости. Стыковочная мишень видна

неустойчиво (в зависимости от условий освещения), метки мишени не видны. Образ поперечного сечения может частично выходить за пределы поля зрения. Стыковочный узел, как правило, виден целиком.



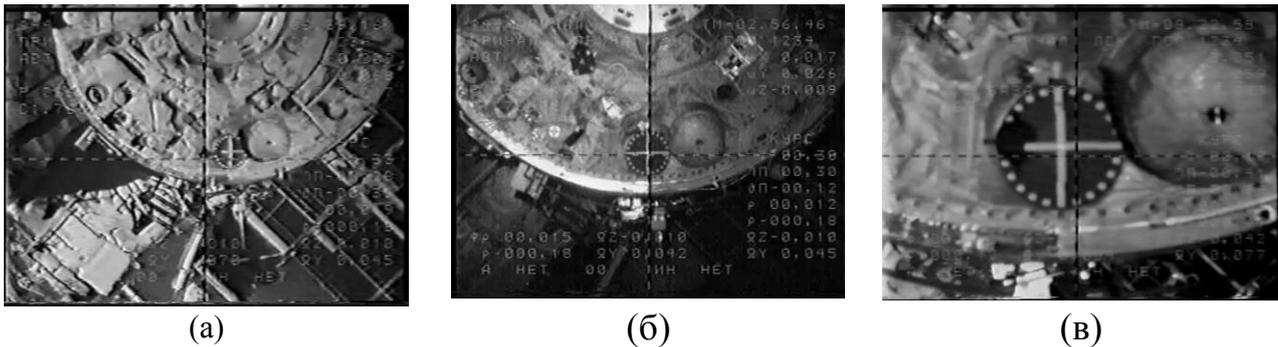
*Рис. 3.* Изображения МКС, полученные в процессе различных сеансов стыковки, типичные при дальности корабля в интервале от 45 м до 25 м.  
(а) Дальность 45 м; (б) 35 м; (в) 30 м.

Дальность от 25 м до 15 м (рис. 4): образ поперечного сечения выходит за пределы поля зрения ТВ-камеры. Стыковочный узел виден устойчиво, но его образ может выходить за пределы поля зрения (не более чем на 50% от площади этого образа). В этом интервале дальностей становится видна стыковочная мишень. На образе мишени присутствует перекрестие, но угловые метки различимы неустойчиво (их образы при некоторых условиях освещения сливаются с фоном вокруг стыковочной мишени и между собой).



*Рис. 4.* Изображения МКС, полученные в процессе различных сеансов стыковки, типичные при дальности корабля в интервале от 25 м до 15 м.  
(а) Дальность 25 м; (б) 22 м; (в) 20 м.

Дальность от 15 м до 0 м (до момента стыковки, рис. 5). На таких расстояниях между КК и станцией на обрабатываемых изображениях становятся хорошо видны стыковочная мишень и ее отдельные части – перекрестие и угловые метки. Образ стыковочного узла выходит за пределы поля зрения, в окрестности внешнего края станции наблюдается большое количество связанных областей различной яркости и контрастные контуры.



*Рис. 5.* Изображения МКС, полученные в процессе различных сеансов стыковки, типичные при дальности корабля в интервале от 15 м до 0 м.  
(а) Дальность 15 м; (б) 10 м; (в) 1 м.

С учетом выделения четырех различных классов обрабатываемых изображений было выполнено проектирование высокоуровневого алгоритма обработки изображений для ПО СТЗ. Этот алгоритм был спроектирован в виде композиции четырех алгоритмов, выполняющих обработку изображений различных классов. Все эти алгоритмы оформлены в виде классов, унаследованных от базового класса «Высокоуровневый алгоритм обработки зрительных данных» каркаса ПО СТЗ. Разбиение высокоуровневого алгоритма обработки изображений на классы-алгоритмические сущности привносит в ПО СТЗ понятие «текущий алгоритм обработки». В зависимости от определенных условий (подход КК на заданное расстояние) происходит переключение ПО СТЗ на соответствующий алгоритм обработки.

Высокоуровневый алгоритм обработки зрительных данных в рассматриваемой СТЗ должен обеспечивать обнаружение объектов интереса с учетом возможности запуска автоматической обработки в произвольный момент времени в процессе прохождения КК прослеживаемого интервала. Для обеспечения этого в каждом из четырех алгоритмов для различных интервалов расстояний реализованы собственные варианты поиска «сверху-вниз» и «снизу-вверх». Таким образом, эти алгоритмы могут применяться как для первоначального обнаружения, так и для прослеживания объекта интереса. В зависимости от интервала расстояний, на который рассчитан алгоритм обработки, в нем производится обнаружение образов различных частей станции:

- Дальность от 100 м до 45 м. Выполняется поиск поперечного сечения станции в следующем порядке: сначала обнаруживается внешний контур образа торца служебного модуля, затем эта гипотеза может быть подтверждена посредством нахождения образа стыковочного узла или солнечных батарей.
- Дальность от 45 м до 25 м. Выполняется поиск поперечного сечения станции: сначала выполняется поиск образа стыковочного узла, затем

размеры и местоположения поперечного сечения уточняются посредством поиска внешнего кругового контура станции.

- Дальность от 25 м до 15 м. Производится поиск стыковочной мишени, затем выполняется попытка поиска меток (в случае успешного поиска обнаруженные метки используются для подтверждения гипотезы обнаружения мишени и уточнения размеров ее образа) и поиск сегмента окружности поперечного сечения станции.
- Дальность от 15 м до 0 м. Выполняется поиск мишени: сначала обнаруживается перекрестие, затем метки мишени. Контур поперечного сечения станции не обнаруживается, т.к. на данном расстоянии в окрестности внешнего кругового контура присутствует большое количество контрастных контурных элементов, затрудняющих его надежное обнаружение.

Обработка зрительных данных в рассматриваемой СТЗ осложняется наличием посторонних искусственных образов на изображениях: крестообразного визира и символьных данных, представляющих данные определения относительного движения КК и МКС с помощью существующей радиолокационной системы слежения. Центр визира в процессе стыковки позиционируется в центр стыковочной мишени. Таким образом, искомый образ на изображении искажается, и это обстоятельство требует учета в алгоритмах его обнаружения.

Кроме наличия образа визира, особенностью обрабатываемых изображений является изменчивость условий освещения. Можно выделить два типа сеансов сближения и стыковки, условия наблюдения в которых существенно отличаются: при солнечном освещении МКС («дневная стыковка») и при освещении МКС прожектором, установленным на КК («ночная стыковка», рис. 2(в), 4(в), 5(б)). В случае ночной стыковки снижается контраст внешнего контура поперечного сечения станции по отношению к фону. На изображениях могут появляться области, образованные зеркальным отражением при освещении прожектором. При появлении этих областей в окрестности внешнего контура станции (рис. 4(в)) они могут приводить к формированию ложных гипотез о наличии сегмента внешнего контура поперечного сечения станции.

В целом, анализ тестовых видеопоследовательностей в данной задаче позволил установить следующие факты:

- наличие двух объектов интереса на обрабатываемых изображениях – поперечного сечения станции и стыковочной мишени;
- наличие четырех различных классов изображений, каждый из которых может быть охарактеризован однотипным набором признаков, образующих искомые образы объектов интереса;
- присутствие на изображениях помех в виде искусственных образов (визира и символьной информации) и шумовых помех, в частности

связанных с помехами коммуникационного канала (например, рис. 4(б));

- изменчивость условий освещения (проявляется на изображениях в виде разрывности областей однородной яркости, образующих образы объектов интереса и снижению контраста контуров этих областей с окружающим фоном).

Перечисленные факты, совместно с геометрическими характеристиками объектов интереса, были использованы при проектировании высокоуровневых алгоритмов обработки зрительных данных в рассматриваемой СТЗ.

## **Инструментальные средства для визуализации графов**

При выборе инструмента для визуализации семантической структуры данных требуется решить несколько проблем. Необходимо выбрать структуру данных, язык и формат для спецификации структуры, а также инструмент для возможной визуализации описаний. Кроме того, учитывая клиентоориентированность задачи, необходимо, чтобы выбранный язык был удобочитаемым и позволял оперативно корректировать сформированные описания. По результатам анализа известных программных систем, были выбраны язык описания графов DOT [5] и средство визуализации GraphViz [5]. Язык был выбран благодаря своей простоте, удобочитаемости, широкому распространению и обширной поддержке средствами визуализации. Средство визуализации GraphViz было выбрано как надёжное, стабильное, поддерживаемое крупными компаниями (такими как AT&T), с большим сообществом, наличием библиотеки (позволяющей встраивать средство в свою программу) и с открытым исходным кодом.

В качестве примера рассмотрим пример описания видеопоследовательности для системы компьютерного зрения, предназначенной для автоматизации мониторинга процесса сближения и стыковки КК и МКС, а также часть графа, построенного на основе анализа этого описания.

*«Дальность от 100 м до 45 м. На изображениях заметны солнечные батареи и поперечное сечение станции. Внешний контур стыковочного узла виден неустойчиво, в зависимости от условий освещения при выполнении стыковки. В центральной части образа стыковочного узла присутствует темная область. Стыковочная мишень практически не видна».*



Рис. 6. Графовое представление одного из требований для ПО СТЗ слежения за процессом сближения и стыковки КК и МКС.

## Программные продукты и библиотеки визуализации графов

Применительно к задаче управления требованиями к ПО СТЗ был рассмотрен ряд доступных средств визуализации графов. В рамках данной работы рассматривались библиотеки и программы для визуализации графов в соответствии с лицензией, позволяющей свободное изучение исходных текстов и их модификацию, свободное использование в исследовательских и научных целях. Кроме таких программных продуктов, имеются также известные проприетарные и несвободные продукты, в частности, Wolfram Mathematica, Tom Sawyer Software, yEd, OmniGraffle, TouchGraph, Starlight. Некоторые из перечисленных продуктов, несмотря на проприетарность, не являются платными, однако невозможность изучения внутренней структуры и самостоятельного изменения данного ПО не позволяет рассматривать его наравне с открытыми программами.

Разработкой подобных программных продуктов занимаются и крупные корпорации (такие как Microsoft, AT&T), и отдельные разработчики. Среди готовых решений можно найти библиотеки и программы, написанные на разных языках программирования (Java, JavaScript, C, C#, C++), на разных платформах (Java, .NET) и работающие в разных операционных системах (Windows, OS X, Linux).

Перечислим основные приложения для визуализации данных с открытым исходным кодом (в скобках указан разработчик): *BioFabric* (Institute for Systems Biology), *Cytoscape* (Institute for Systems Biology), *Microsoft Automatic Graph Layout* (Lev Nachmanson, Microsoft Research), *Tulip* (David Auber, Patrick Mary, University of Bordeaux), *PGF* (Till Tantau, Christian Feuersaenger), *Gephi* (University of Technology of Compiègne), *Graphviz* (AT&T Labs Research), *библиотеку VizierFX* (Mark Andrew Goetz), *Canviz*, *Viz.js*, *Grappa*, *Dracula Graph Library*, *GUESS* (Eytan Adar), *GVF* (Scott Marshall), *JUNG* (Joshua O'Madadhain, Danyel Fisher, Tom Nelson), *SemaSpace*, *Walrus* (University of California's San Diego Supercomputer Center), *Open Graph Drawing Framework* (Carsten Gutwenger, Markus Chimani, Karsten Klein), *Hamiltonian Graph Builder* (Christopher Manning), *Pajek* (Vladimir Batagelj, Andrej Mrvar).

Стоит отметить особенности некоторых из перечисленных программных продуктов. BioFabric, изначально разрабатывавшийся для нематематических целей, представляет граф в виде горизонтальных и вертикальных линий, где горизонтальные линии – это вершины, а вертикальные – рёбра.

## Языки описания графов

Естественным при выборе средства визуализации является вопрос о доступных способах задания графа и их количестве. Программные продукты и библиотеки, упомянутые выше, различаются по количеству и качеству поддержки форматов файлов с описанием графов. Рассмотрим основные языки описания графов и соответствующие им форматы.

Рассмотрим форматы описания (задания) графов: Newick format [6], TGF [7], DGML [8], GML [10], XGMML [11], GraphML [12], GXL [13], GEXF, LCF notation, DOT (graph description language) [14], DotML.

Newick tree format (Newick notation, New Hampshire tree format) — язык представления деревьев с длиной рёбер, задаваемой с помощью круглых скобок и запятых. Формат был принят как обобщение формата, разработанного в 1984 году для первой программы, визуализировавшей деревья, – PHYLIP.

TGF (Trivial Graph Format) предоставляет функционал для описания непосредственно структуры графа, позволяя назначать вершинам и рёбрам по одному наименованию. В формате не предусмотрена возможность указать дополнительно положение, позицию в пространстве и внешний вид элементов. Не поддерживаются вложенные графы. В проприетарном графическом редакторе yEd и проприетарном же пакете Wolfram Mathematica, упомянутом выше, поддерживается формат TGF.

LCF (LCF-notation) – система обозначений в комбинаторной математике для представления кубических гамильтоновых графов. LCF-нотацию нельзя рассматривать как полноценный специализированный формат файлов или язык задания графов, ввиду специализации только на определённом виде графов. Тем не менее, существует возможность получить визуализацию графа, заданного с помощью LCF. Описание гамильтоновых графов с использованием LCF-нотации и получение визуализации графов, заданных с помощью LCF, можно выполнить с помощью онлайн-программы Hamiltonian Graph Builder.

Graph6 — формат задания графов, предложенный Brendan McKay. Формат использует только печатаемые ASCII-символы, иначе сжимает матрицу смежности в последние 6 бит символа. Полученная запись графа не является удобочитаемой, хотя все полученные символы гарантированно могут быть напечатаны. Для чтения данного формата необходима специальная программа.

Rajek – формат задания больших сетей, состоящих из миллионов вершин, использующийся в одноимённой программе.

DGML (Directed Graph Markup Language) – XML-подобный формат файлов для разметки ориентированных графов. Достаточно узкоспециализированный язык, относительно молодой. Впервые появился и используется в Microsoft

Visual Studio 2010. Используется для описания и последующей визуализации отношений и зависимостей в программном коде.

GML (Graph Modeling Language or Graph Meta Language) – иерархический файловый формат для описания графов. Многие приложения способны работать с GML. Из упомянутых выше программных продуктов формат поддерживают Cytoscape, Gephi, GraphTool, Open Graph Drawing Framework, Tulip, yEd.

XGMML (eXtensible Graph Markup and Modeling Language) – язык описания графов, являющийся XML-подобным расширением Graph Modeling Language. Язык спроектирован таким образом, что имеется возможность однозначного перевода графа из формата GML в XGMML. Изначально создавался для системы WWWPAL, которая обеспечивает визуальное представление сайта в виде графа. Из перечисленных выше программ, поддержку XGMML обеспечивает пакет Cytoscape.

GraphML (Graph Markup Language) – формат файлов для описания графов, основанный на синтаксисе XML. Форматом обеспечивается поддержка описания всех видов графов (ориентированных, неориентированных, смешанных, гиперграфов) и необходимых вспомогательных атрибутов. Из упомянутых выше программ частичная поддержка GraphML имеется в Gephi, а в проприетарном yEd формат используется в качестве основного.

GXL (Graph eXchange Language) проектировался как стандартный формат обмена графов. Синтаксис GXL задан с помощью XML document type definition и является XML-подобным. Так как в виде графа можно представить и другие структуры данных (такие как таблицы, списки и деревья), GXL предоставляет отличные возможности интероперабельности между различными информационными хранилищами и приложениями. К сожалению, проект и стандарт в последнее время не развиваются, так как последняя спецификация (версия 1.1), согласно официальному сайту, датируется 2002-м годом.

DOT: GraphViz, OmniGraffle, VizierFX, ZGRViewer, Microsoft Automatic Graph Layout (online-utility is available <http://rise4fun.com/agl>)  
Неориентированный граф в языке DOT можно описать, используя следующую конструкцию:

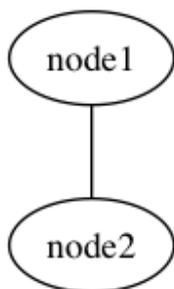
```
graph name
{
    node1 [label = "node1"];
    node2 [shape = plaintext];
    node1 -- node2;
}
```

Описания вершин в примере выше не являются обязательными, и их можно опустить. В описании графа, а также квадратных скобках справа от названия вершины или ребра можно указывать различные модификаторы. В простейшем случае граф может быть построен по такому описанию (рис. 7(a)):

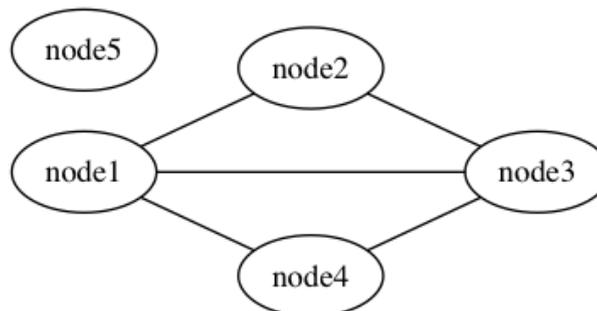
```
graph simple
{
    node1 -- node2;
}
```

Ниже приведен пример описания графа с 5 вершинами (рис. 7(б)):

```
graph example1
{
  rankdir=LR;
  splines=false;
  node1 -- node2 -- node3 -- node4 -- node1;
  node1 -- node3;
  node5;
  { rank=same node2, node4 }
}
```



(a)



(б)

Рис.7. Примеры графов, построенных по описанию на языке DOT с помощью программы GraphViz.

Как видно из представленных описаний, форматы описания графов можно разделить на XML-подобные и свободные форматы. Явное преимущество первых – наличие большого количества библиотек по автоматической обработке XML-подобных форматов, что значительно облегчает их разбор, анализ и построение, упрощает разработку прикладного ПО. Но для восприятия человеком XML в ряде случаев оказывается достаточно сложным. Большое количество вспомогательных конструкций и служебной информации затрудняют чтение файлов с описанием в формате XML и не позволяют неспециалисту быстро включиться в работу по чтению, анализу и описанию графов.

## Заключение

Для актуальной задачи формулировки требований в процессе разработки систем компьютерного зрения реального времени была спроектирована и разработана программа для разбора и анализа текстовых описаний видеопоследовательностей. Трансляция видеопоследовательностей и их текстовых описаний в графовые структуры данных позволяет упростить дальнейшую разработку алгоритмов обработки зрительных данных и тестирование работоспособности системы компьютерного зрения.

В работе перечислены программные продукты и библиотеки, позволяющие решать задачи визуализации данных в виде графов. Для дальнейших исследований и разработок был выбран язык DOT, как достаточно простой и

выразительный для настройки получаемого графа. В то же время DOT, в отличие от Trivial Graph Format, предоставляет средства дополнительного описания и выразительности, значительно расширяя выразительные и описательные способности и позволяя автоматически генерировать отличные наглядные изображения.

## Библиографический список

1. David Ferucci, Eric Brown. Building Watson: An Overview of the DeepQA Project // AI Magazine – 2010 – Vol.31, No 3
2. 830 IEEE Recommended Practice for Software Requirements Specifications. - URL: <http://ieeexplore.ieee.org/servlet/opac?punumber=5841>
3. 1233 IEEE Guide for Developing System Requirements Specifications. – URL: <http://ieeexplore.ieee.org/servlet/opac?punumber=5982>
4. 29148 Systems and software engineering -- Life cycle processes --Requirements engineering. 2011  
URL: <http://ieeexplore.ieee.org/servlet/opac?punumber=6146377>
5. Gansner E. R., North S.C. An open graph visualization system and its applications to software engineering // Software-Practice and Experience – 1999 – 00(S1), 1–5
6. Язык описания графов Newick Tree Format.  
URL: <http://evolution.genetics.washington.edu/phylip/newicktree.html>
7. Язык описания графов TGF.  
URL: <http://docs.yworks.com/yfiles/doc/developers-guide/tgf.html>
8. Язык описания графов LCF.  
URL: <http://mathworld.wolfram.com/LCFNotation.html>
9. Язык описания графов DGML.  
URL: <http://schemas.microsoft.com/vs/2009/dgml/>
10. Язык описания графов GML.  
URL: <http://www.fim.uni-passau.de/index.php?id=17297&L=1>
11. Язык описания графов XGMML.  
URL: [http://cgi7.cs.rpi.edu/research/groups/pb/punin/public\\_html/XGMML/](http://cgi7.cs.rpi.edu/research/groups/pb/punin/public_html/XGMML/)
12. Язык описания графов GraphML. URL: <http://graphml.graphdrawing.org/>
13. Язык описания графов GXL. URL: <http://www.gupro.de/GXL/>
14. Язык описания графов DOT.  
URL: <http://www.graphviz.org/content/dot-language>

## Оглавление

Введение .....	3
Автоматическая обработка текстовых данных .....	3
Пример требований к программному обеспечению СТЗ реального времени .....	4
Инструментальные средства для визуализации графов .....	10
Программные продукты и библиотеки визуализации графов .....	11
Языки описания графов .....	12
Заключение.....	14
Библиографический список.....	15