

**Российская ассоциация искусственного интеллекта  
Высшая школа информационных технологий  
и информационных систем  
Казанского федерального университета**

---



**БТС  
ИИ**

**ЧЕТВЕРТЫЙ ВСЕРОССИЙСКИЙ  
НАУЧНО-ПРАКТИЧЕСКИЙ СЕМИНАР  
«БЕСПИЛОТНЫЕ ТРАНСПОРТНЫЕ СРЕДСТВА  
С ЭЛЕМЕНТАМИ ИСКУССТВЕННОГО  
ИНТЕЛЛЕКТА»**

**5-6 октября 2017  
Казань, Россия**

**Труды семинара**

*Осуществлено при финансовой поддержке РФФИ и Правительства  
Республики Татарстан в рамках проекта № 17-41-161035*

**К а з а н ь  
Центр инновационных технологий  
2 0 1 7**

УДК (629.052.9+004.8)(470.41-25)(082)  
ББК 39.175я431+32.813я431  
Ч-52

**Четвертый** Всероссийский научно-практический семинар  
**Ч-52** «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2017, 5-6 октября 2017 г., г. Казань, Республика Татарстан, Россия): тр. семинара / под ред. Е.А. Магида, В.Е. Павловского, К.С. Яковлева. – Казань: Центр инновационных технологий, 2017. – 240 с.  
ISBN 978-5-93962-833-4

В сборник включены тексты работ, представленные на четвертом Всероссийском научно-практическом семинаре «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2017) 5-6 октября 2017 г.

*Официальный сайт семинара – [www.ai-uv.ru](http://www.ai-uv.ru)*

ISBN 978-5-93962-833-4

© Коллектив авторов, 2017  
© Российская ассоциация  
искусственного интеллекта, 2017  
© Центр инновационных технологий  
(оформление), 2017

## О семинаре

В настоящее время наблюдается существенное повышение интереса исследователей и разработчиков к созданию беспилотных транспортных средств различного типа и назначения, способных к автономному решению высокоуровневых задач в динамических, непрогнозируемых средах. Создание подобных средств невозможно без интеграции усилий специалистов в различных областях науки: механики, теории управления, теории передачи информации, компьютерной графики, распознавания образов, искусственного интеллекта, когнитивных наук и многих других. Одним из механизмов указанной интеграции является проведение семинара «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ), в ходе которого исследователи различных специализаций имеют возможность обмениваться опытом решения актуальных проблем, связанных с созданием беспилотных транспортных средств нового поколения – таких средств, которые могли бы быть названы интеллектуальными.

Первый семинар БТС-ИИ состоялся 24 сентября 2014 года в Казани в рамках Четырнадцатой национальной конференции по искусственному интеллекту с международным участием (КИИ-2014). В 2015 году семинар прошел в Санкт-Петербурге в рамках Международной научно-технической конференции «Экстремальная Робототехника». В 2016 году семинар БТС-ИИ впервые стал самостоятельным мероприятием, которое прошло в университете Иннополис нового российского IT-города Иннополис. Программа семинара состояла из выступлений 2 приглашенных и 17 регулярных докладчиков, и круглого стола Фонда перспективных исследований.

В этом году семинар БТС-ИИ состоится в Высшей школе информационных технологий и информационных систем Казанского (Приволжского) федерального университета. Программа семинара включает выступление 2 приглашенных и 22 регулярных докладчиков, и проведение круглого стола Российского фонда фундаментальных исследований (РФФИ). В этом году к семинару также был проявлен интерес со стороны как официальных фондов, так и частных компаний – финансовую поддержку семинару оказали Российский фонд фундаментальных исследований, Правительство Республики Татарстан, компании КУКА и «Сервсила».

Желаем успехов участникам семинара и надеемся на плодотворную дискуссию в ходе работы!

## **Организатор**

Российская ассоциация искусственного интеллекта

Высшая школа информационных технологий и информационных систем  
Казанского федерального университета

## **Финансовая поддержка**

- Российский фонд фундаментальных исследований и Правительство Республики Татарстан в рамках проекта № 17-41-161035
- КУКА Роботикс Рус
- Компания Сервосила

## **Программный комитет**

**В.Е. Павловский (сопредседатель)**, доктор физико-математических наук, профессор, главный научный сотрудник Института прикладной математики им. М.В. Келдыша РАН, член научного совета Российской ассоциации искусственного интеллекта.

**К.С. Яковлев (сопредседатель)**, кандидат физико-математических наук, ведущий научный сотрудник Федерального исследовательского центра "Информатика и управление" Российской академии наук, член научного совета Российской ассоциации искусственного интеллекта.

**И.М. Афанасьев**, кандидат технических наук, доцент Института Робототехники, Университет Иннополис.

**Д.А. Добрынин**, кандидат технических наук, старший научный сотрудник Федерального исследовательского центра "Информатика и управление" Российской академии наук, член Российской ассоциации искусственного интеллекта.

**В.Э. Карпов**, кандидат технических наук, доцент, руководитель Отделения нейрокогнитивных наук и интеллектуальных систем НИЦ "Курчатовский институт", вице-президент Российской ассоциации искусственного интеллекта.

**Н.В. Ким**, кандидат технических наук, профессор, лауреат премии правительства РФ в области образования, профессор кафедры 704 факультета №7 "Робототехнические и интеллектуальные системы" Московского авиационного института.

**Е.А. Магид**, PhD, профессор, заведующий кафедрой "Интеллектуальная Робототехника" Высшей школы информационных технологий и информационных систем, Казанский Федеральный Университет.



**С.Б. Ткачев**, доктор физико-математических наук, профессор, лауреат премии правительства РФ в области науки и техники, профессор кафедры "Математическое моделирование" МГТУ им. Н.Э. Баумана.

**Официальный сайт семинара** – <http://2017.ai-uv.ru/>

## Спонсоры семинара



## Правительство Республики Татарстан

Российский фонд фундаментальных исследований совместно с Правительством Республики Татарстан в рамках соглашения №17-41-161035 оказали финансовую поддержку для реализации проекта организации IV Всероссийского научно-практического семинара «Беспилотные транспортные средства с элементами искусственного интеллекта».

# УЧЕБНЫЙ СТЕНД НА БАЗЕ КУКА KR AGILUS

Предназначен для образовательных целей:

- ▼ в средне-специальных учреждениях
- ▼ в высших учебных заведениях
- ▼ в центрах повышения квалификации
- ▼ в центрах дополнительного и профессионального образования.

Специализированная оснастка и комплект сменных инструментов, поставляемые в составе рабочего основания, позволяют отработать навыки управления и программирования промышленного робота.

Позволяет изучать устройство, принципы управления и программирования, механику промышленных роботов на примере промышленного робота-манипулятора Kuka KR Agilus.

Промышленный манипулятор с шестью степенями свободы, установленный на специализированное рабочее основание.

Манипулятор Kuka KR Agilus рекомендован для использования при подготовке специалистов к этапам соревнований WorldSkills Russia (WSR) по направлению «Промышленная робототехника».

**KUKA** 

# **Приглашенные доклады**

## Роботизация поисково-спасательных операций

Е.А. Магид (*magid@it.kfu.ru*)

Казанский федеральный университет, г. Казань

Сегодня одной из самых актуальных задач робототехники является задача разработки поисково-спасательных роботов, которые увеличивают безопасность спасательных команд и одновременно существенно расширяют технические возможности команды. Такие роботы многофункциональны и могут применяться не только спасателями МЧС во время обследования района бедствия и поиска пострадавших, но и для изучения поверхности планет, обследования подземных коммуникаций, лабиринтов и шахт, разведывательных и антитеррористических миссий, военных и полицейских операций, оценки обстановки на зараженных объектах, дистанционной ликвидации взрывных устройств и пр.

Во время поисково-спасательной операции робот дислоцируется в заданном районе бедствия для выполнения миссии и управляется оператором, находящимся на безопасном расстоянии, дистанционно по беспроводной связи или же с использованием кабельного соединения. При этом, несмотря на то, что основная нагрузка по управлению и принятие решений остаются прерогативами человека-оператора, часть базового функционала робота должна быть в обязательном порядке автоматизирована для снижения физической и психоэмоциональной нагрузки на оператора, увеличения скорости и качества выполнения технических подзадач.

В докладе будут рассмотрены принципы и методы проведения поисково-спасательных работ при помощи робототехнических систем различных типов. Также будут представлено несколько актуальных научно-исследовательских проектов лаборатории интеллектуальных робототехнических систем (ЛИРС) Казанского федерального университета, целью которых является повышение безопасности и эффективности поисково-спасательных работ. В качестве основной мобильной платформы для прикладных исследований нами используется гусеничный робот повышенной проходимости российского производства «Сервосила Инженер». Разработка программного обеспечения для робота осуществляется под стандарты робототехнической операционной системы ROS, в то время как для прототипирования алгоритмов используется среда Matlab, а для реалистических симуляций с визуализацией - среда Gazebo.

В ближайшие три года команда ЛИРС рассчитывает создать первый прототип робота-помощника для нужд МЧС с перспективой дальнейшего широкого внедрения в Республике Татарстан в течение пяти-семи лет.

## Мемристорные эмоциональные вычисления

М.О. Таланов (*max.talanov@gmail.com*)  
Казанский федеральный университет, г. Казань

Некоторое время назад, мы начали проект, задаваясь вопросом "как заставить машину почувствовать эмоции". После 3-х лет работы мы смогли построить био-реалистичную модель основных психо-эмоциональных состояний: страха, отвращения, унижения, радости, интереса, удивления, злости и грусти основываясь на модели Hugo L'ovheim (2012).

Мы расширили данную модель, добавив в нее параметры вычислительной системы: потребление вычислительных ресурсов, распределение вычислительных ресурсов, потребление памяти и связанность ячеек памяти. Позднее мы реализовали эту модель, используя нейросимулятор NEST используя подсистемы дофамина, серотонина и норадреналина крысиного мозга и смогли био-реалистично реализовать состояния подобные страху и отвращению. К этому моменту нам стало понятно, что нам нужен новый тип аппаратного обеспечения для интеграции с робототехнической системой реального времени. Текущая симуляция использует до 10 нод вычислительного кластера КФУ и производит вычисления одной секунды симуляции в течении недели. К этому моменту мы начали сотрудничать с "отцом органических мемристоров" Виктором Ерохиным. В результате родилась идея использовать органические мемристоры как электронные аналоги синапсов в ингибирующих и нейромодулирующих компонентах STDP посредством электронной схемы.

Я представлю результаты работ по воспроизведению биоинспирированного обучения STDP как по возбуждению, так и по ингибированию. Также будут представлены результаты как симуляционной валидации, так и физического прототипирования нейромодулирующего мемристорного нейрона. Данные исследования произведены впервые в мире и не имеют аналогов.

## Содержание

***А.К. Буйвал, М.А. Гавриленков***

Реализация алгоритма визуальной локализации беспилотного летательного аппарата на основе известной 3D модели окружения с использованием CNN-сегментации 14

***В.А. Михайлов, О.Г. Пилипенко***

Распознавание основных объектов инфраструктуры городской местности при помощи БПЛА и нейросети U-net 25

***А.В. Бокковой***

Метод решения задачи одновременного картирования и локализации по видеопотоку и данным инерциальной навигационной системы для малых БПЛА 36

***П.С. Сорокоумов***

Система компьютерного зрения для распознавания элементов управления автомобилем роботом-водителем 46

***А.Д. Московский***

Графы как инструмент для решения задачи распознавания сцен 56

***К.С. Шабалина, А.Г. Сагитов, Е.А. Магид***

Сравнение систем координатных меток для калибровки камер мобильного робота в условиях перекрытий 65

***А.А. Закиев, Р.О. Лавренов, Е.А. Магид***

Программный инструмент для создания 3D-карт в Gazebo на основе произвольных изображений или данных лазерного сканирования 76

***А.Г. Сагитов, Ю.А. Герасимов, Е.А. Магид***

Моделирование трехосного подвеса камеры квадрокоптера DJI и радиоуправления в среде ROS / Gazebo 87

***Р.О. Лавренов, Е.А. Магид***

Поиск маршрута для наземного робота: модифицированный алгоритм планирования на основе сплайнов 96

- А.А. Андрейчук, К.С. Яковлев**  
 Планирование траектории на плоскости с учетом размера агента (мобильного робота, беспилотного транспортного средства) 107
- Д.А. Добрынин**  
 Задача обучения движению по траектории беспилотного транспортного средства с использованием ДСМ-метода 118
- Н.А. Михайлов**  
 Планирование маршрута поиска для автономного беспилотного летательного аппарата с использованием энтропийного подхода 126
- С.Л. Зенкевич, Хуа Чжу, Цзяньвень Хо**  
 Движение группы мобильных роботов в строю типа "конвой" — теория, моделирование и эксперимент 136
- А.В. Леонард, Е.С. Брискин, И.А. Горбов**  
 Планирование законов движения наземным БТС на основе целевой функции, задаваемой оператором 148
- Н.В. Ким, Н.Е. Бодунков**  
 Формирование семантических описаний для решения целевых задач автономными беспилотными летательными аппаратами 158
- В.Н. Жидков, Н.В. Ким**  
 Поиск согласованных решений человеком и роботом. Проблема манипулирования 165
- Г.В. Ройзензон, В.Э. Карпов, В.Е. Павловский, В.Б. Бритков**  
 Построение многомерного классификатора интеллектуальных роботов 175
- А.Л. Ронжин, Д.К. Ву, В.В. Нгуен, О.Я. Соленая**  
 Концептуальная и алгоритмические модели совместного функционирования роботизированной платформы и набора БЛА при выполнении аграрных операций 183
- А.А. Кулинич**  
 Метод построения семиотической среды функционирования группой интеллектуальных агентов 193



***Г.А. Киселев, А.И. Панов***

Планирование действий коалицией агентов: коммуникационный аспект 204

***В.В. Воробьев***

Коммуникационный аспект задачи исследования области группой роботов с локальным взаимодействием 216

***А.Р. Гамаюнов, Е.М. Притоцкий, П.К. Герасимов, Д.А. Егоров***

Удалённая коммуникация множества пользователей с мультиагентной распределённой системой на основе мобильных автономных БТС 226

УДК 004.932.2

## РЕАЛИЗАЦИЯ АЛГОРИТМА ВИЗУАЛЬНОЙ ЛОКАЛИЗАЦИИ БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА НА ОСНОВЕ ИЗВЕСТНОЙ 3D МОДЕЛИ ОКРУЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ CNN- СЕГМЕНТАЦИИ.

А.К. Буйвал (*alexbuyval@gmail.com*)

М.А. Гавриленков (*gavrilentov@umlab.ru*)

Брянский Государственный Технический Университет,  
Брянск

**Аннотация.** В статье рассматривается новый подход к визуальной локализации в помещении, основанный на использовании CNN-сегментации изображений для улучшения ранее разработанного алгоритма локализации. Авторы предлагают ряд нововведений для устранения таких проблем, как шум, создаваемый нестатическими объектами или группами непредвиденных объектов в помещении. Информацию о таких объектах в данной работе считается избыточной и исключается из алгоритма локализации. В данной статье рассматривается подход исключения избыточной информации с видеокамеры для локализации в пространстве посредством использования сверточных нейронных сетей для преобразования изображения в сегментированную карту поверхности, на которой хорошо прослеживаются отличительные признаки (плоскости и грани) изображения. В основе алгоритма лежит определение границ объектов на изображении с камеры и сопоставлении этих границ с моделью.

**Ключевые слова:** визуальная локализация, нейронные сети, ROS, фильтр частиц, SegNet.

### Введение

Проблема локализации мобильного робота остается актуальной и по сей день. Локализация мобильного робота, а в частности БПЛА является критически важной функцией для функционирования робота внутри помещения. В отличие от роботов, которые предназначены для выполнения задач на открытой местности, роботы функционирующие внутри помещений, не могут воспользоваться датчиками глобальной навигации

(GPS/ГЛОНАСС). Для выполнения локализации, построения маршрутов, а также принятия решений очень часто используются инерциальные датчики, лазерные дальномеры или данные с видеокамеры.

Одним из прогрессивных подходов к решению задачи локализации является использование информации, полученной с бортовой камеры БПЛА и применении фильтра частиц. Данный подход основывается на сравнении изображения, полученного с камеры, со смоделированным изображением, полученным на основе 3D модели помещения. Этот подход во многом превосходит альтернативные решения, но, к сожалению, не лишен недостатков. Весомым недостатком становится избыточность получаемой с камеры информации.

В данной статье предлагается рассмотреть идею предобработки кадров с видеокамеры для дальнейшего использования в алгоритме локализации.

## **1. Базовый алгоритм локализации**

В ранних работах [Буйвал, 2015, 2017] описывается алгоритм визуальной локализации с использованием граней изображений как основного характерного визуального признака. В данных работах приводится способ оценки схожести двух изображений (изображения с реальной камеры и изображения с камеры модели), как основного критерия оценки гипотез о местоположении робота в пространстве. Для обработки данных о схожести изображений, а также данных с других доступных датчиков используется алгоритм локализации, основанный на множестве частиц (гипотез о местоположении робота) – фильтр частиц (particle filter).

Для того, чтобы оценить вероятность каждой гипотезы (частицы) необходимо оценить близость границ, полученных из изображения с камеры, к границам, полученным из смоделированного изображения, соответствующего гипотезе.

В разработанной системе используется метод ближайших граней. Данный метод не имеет такого недостатка, как учет только совпадающих на гранях пикселей. Т.о. если грани одного и того же конструктивного элемента на изображении с камеры и на смоделированном изображении не совпадают, но проходят близко друг к другу, то это все равно будет учитываться в конечной вероятности, т.к. в данном методе предлагается оценивать близость смоделированной грани к грани на изображении с камеры с помощью набора нормалей, построенных от смоделированной грани.

В рассматриваемом методе выполняются последовательно следующие этапы:

1. Выделение границ на изображении, полученном с камеры.

2. Нахождение прямых в выделенных границах с помощью преобразования Хафа. Полученные прямые используются как исходные данные для вычисления вероятности каждой частицы.
3. Рендеринг изображения согласно вектору состояния частицы.
4. Выделение границ на полученном изображении.
5. Нахождение прямых в выделенных границах с помощью преобразования Хафа.
6. На каждой прямой формируется набор точек с постоянным шагом. Из каждой точки строится нормаль некой предельной длины. Если эта нормаль пересекается с какой-либо прямой из полученных на основе изображения с камеры, то длина такой нормали учитывается в общем весе рассматриваемой прямой по следующей формуле:

$$g(d) = \exp\left(-\frac{d^2}{2\sigma^2}\right), \quad (1)$$

где  $d$  – длина нормали,  $\sigma$  – параметр определяющий вес нормали в зависимости от длины нормали (позволяет либо усиливать влияние длинных нормалей, либо уменьшать).

7. Для каждой прямой вычисляется общий вес путем суммирования всех весов нормалей по следующей формуле:

$$l = \frac{\sum_{i=0}^S g(d_i)}{S}, \quad (2)$$

где  $S$  – общее количество прямых в смоделированном изображении.

8. Вычисляется итоговая вероятность гипотезы на основе объединения весов каждой линии по следующей формуле:

$$W = \alpha \cdot \exp\left(k \frac{\sum_{n=0}^m l_n}{m}\right), \quad (3)$$

где  $m$  – количество прямых,  $\alpha, k$  – параметры учета веса прямых в итоговом результате.

## 2. Сегментация изображений

Подавляющее большинство методов локализации основанных на обработке видеопотока используют так называемые характерные признаки изображения (ребра, грани, ключевые точки и т.д.). Соответственно, для локализации необходимо выделить эти признаки на изображении. Из этого вытекает проблематика избыточной/недостаточной информации по результатам выделения признаков. Зачастую эта разница становится ощутима при перемещении из меньшего помещения в большее, и наоборот, или же, в случаях сильной загруженности помещения. В частности, в работах [Буйвал, 2015, 2017] в качестве характерных признаков выступают ребра. Для нахождения ребер на изображении применяется связка «оператор Кэнни» + «преобразование Хафа». Результатом становится набор предполагаемых ребер изображения, соответствующих заданным

начальным условиям выборки (погрешность, минимальная длина и т.д.). В данной работе используются реализации оператора Кэнни и преобразований Хафа в пакете OpenCV. На рис. 1 представлен случай выявления граней изображения данным способом, но с неудачными начальными значениями. Как видно из рисунка, в тех случаях, когда изображение имеет четкие переходы цветов, и можно выделить непрерывные длинные ребра (левая часть изображения), алгоритм обрабатывает без проблем (красные круги оказались в точности на синих линиях). Тем не менее, в центральной и правой части ребра не были найдены, более того, в правой части изображения были сформированы избыточные ребра из-за изображенного там объекта. Подобные эффекты возникают из-за наличия множеств оттенков одного цвета на изображении.

Долгое время мы работали над различными подходами к улучшению нашего алгоритма локализации. Основной проблемой для нас было то, что мелкие предметы в помещении дают непропорционально большие помехи. Другой проблемой для нас стал выбор параметров инициализации для функции выделения граней. Здесь явно прослеживалась зависимость выбора критериев (порог толерантности, минимальная длина линии) с качеством локализации. К несчастью, золотой середины в выборе параметров найти не удалось. В зависимости от освещенности помещения, а также его загруженности посторонними предметами, выбор параметров определял качество локализации. К примеру, в загруженных предметами помещениях, следует использовать большее значение для параметра минимальной длины линии. В помещениях, где присутствуют отражающие поверхности, или комната наполнена солнечными бликами, следует повысить значение коэффициента толерантности. Отдельным случаем становятся комнаты, где поверхность пола или стен содержат крупные линии (например, крупная плитка на полу, обои с прямым рисунком).

Решением данной проблемы становится использование сверточных нейронных сетей. Одним из вариантов использования CNN является сегментация изображений. Сегментация изображений применяется для выделения отдельных объектов (человек1, человек2, стул1, стол1, стул2), или же для выделения групп объектов (люди, пол, потолок, мебель и т.д.). Оба эти подхода позволяют снизить количество используемых цветов в изображении.

В качестве базовой нейронной сети для модернизированного алгоритма была выбрана нейросеть SegNet [Kendall, 2015; Badrinarayanan, 2015, 2017]. Выбор нейросети обуславливается компромиссным оптимумом между качеством сегментации (82-85%) и скоростью сегментации (0.01-0.02с). На рисунке 2 представлено тоже самое помещение, но обработанное с помощью нейронной сети. Как видно, довольно четко остаются выражены

относительные границы между полом и потолком, а также дверь в левой части изображения.



Рис. 1. Результат поиска границ на изображении и их сопоставления с границами, полученными из смоделированного изображения (синие линии – ребра из моделируемого изображения, зеленые линии и красные круги – найденные на изображении с камеры)

Применение нейронной сети позволило снизить количество используемых цветов от нескольких миллионов до 12 основных. Таким образом отпадает потребность в подборе параметров для применения преобразований Хафа. Теперь достаточно выбрать оптимальные параметры, которые будут применяться во всех операциях.

На рис. 2 также представлены результаты применения связки «оператор Кэнни» + «преобразование Хафа» к сегментируемому изображению. Видно, что количество найденных ребер, которые соответствуют смоделированному изображению, превышает количество ребер на изображении без предварительной сегментации. В процессе сегментации кадра на изображении могут возникать артефакты (правый нижний угол), это обуславливается недостаточным опытом нейронной сети и информационной энтропией. Тем не менее, при достаточном уровне обучения нейронной сети, такие артефакты слабо влияют на локализацию. В первую очередь следует обращать внимание на стык стен и пола – такие границы наиболее четко описывают помещение.

### 3. Реализация модернизированного алгоритма в ROS

Базой для разработки подсистемы локализации выбрана система ROS. Robotic Operation System позволяет разрабатывать масштабируемые приложения, развертываемые на платформе Ubuntu или Debian.

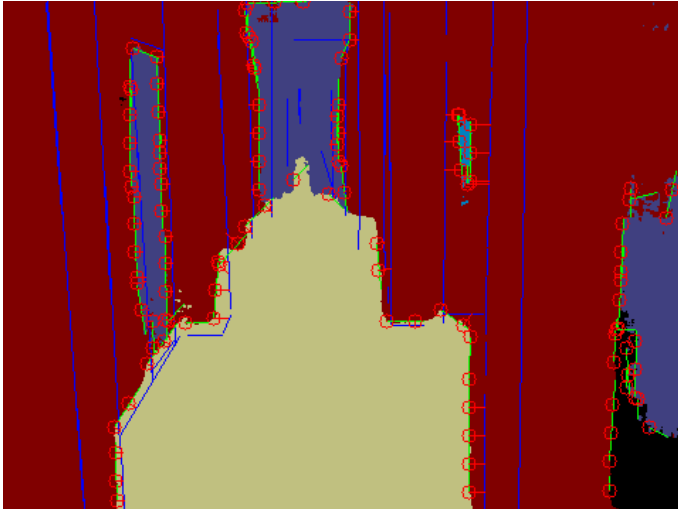


Рис. 2. Результат сегментации изображения с последующим поиском и сравнением найденных граней

На рис. 3 показана схема подсистемы локализации БПЛА без дополнительных модернизаций. Основным модулем (узлом, в терминологии ROS) является `edge_localization_node`. Данный узел получает изображение с камеры БПЛА, после выделения ключевых линий на изображении пересылает их в узел `rendering_node`.

В качестве программной реализации улучшения базового алгоритма выступает новый узел `neironet_adapter_node`. Данный узел служит связью между нейронной сетью и подсистемами ROS. Текущий узел получает от узла `edge_localization_node` исходное изображение, после чего к нему применяется сегментация. Сегментированное изображение возвращается в `edge_localization_node` и базовый алгоритм продолжается [Буйвал, 2017].

### 4. Экспериментальный анализ алгоритма

Для проверки работоспособности системы был проведен следующий эксперимент.

Была создана модель офисного коридора протяженностью 10 метров, протяженная сторона ориентирована по оси Y. В качестве объекта исследования используется БПЛА типа квадрокоптер. Задача состоит в том, чтобы БПЛА пролетел от начала коридора до его конца, используя только визуальные датчики. Во время эксперимента квадрокоптер управлялся в ручном режиме и преимущественно двигался вдоль коридора.

**Целью** эксперимента является определение минимального времени и точности локализации.

В данном эксперименте подсистема работает в режиме «offboard», это означает, что все вычисления происходят на внешнем управляющем компьютере со следующими характеристиками: *ОЗУ: 8Гб; Процессор: Intel Core i7 7700HQ 2.8ГГц; видеокарта: NVIDIA GeForce GTX 1050(m) 2ГБ*. Изображение с камеры квадрокоптера поступало в режиме реального времени на внешний компьютер.

В результате эксперимента были получены числовые значения распределения временной задержки по каждому из этапов локализации.

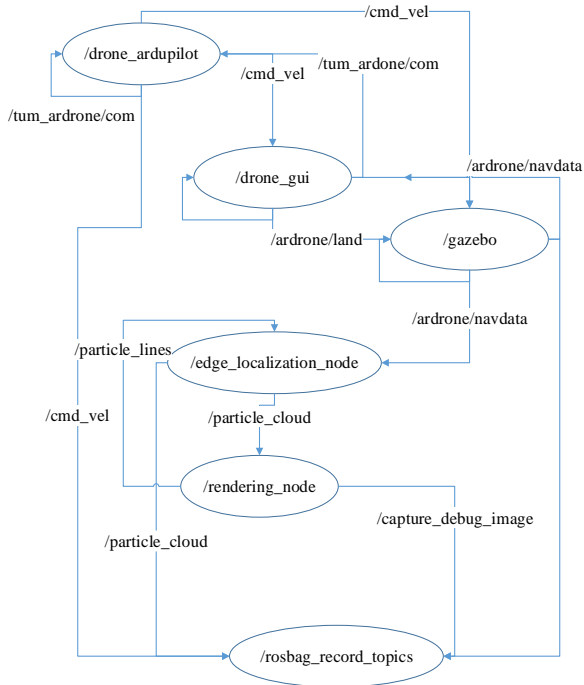


Рис. 3. Схема подсистемы локализации



Как видно из таблицы, наибольшая доля времени уходит именно на сегментацию изображения с помощью нейронной сети. Таким образом, время локализации для одного кадра не превышает 200 мс. В целом, данное время является допустимым для локализации при условии медленного движения БПЛА. Предполагается, что сократить это время возможно за счет увеличения вычислительной мощности графического процессора и использовании более прогрессивных архитектур нейронных сетей.

Таблица 1

Распределение времени работы в модуле сегментации

	Съемка кадра, мс	Изменение размера изображения, мс	Обработка в SegNet, мс	Представление результатов, мс	Суммарный результат, мс
min	4,43	1,35	165,74	4,51	176,05
avr	4,47	1,54	167,10	4,23	177,36
max	4,50	1,74	168,47	3,94	178,67

На рис. 4,5 представлено сравнение рассчитанных и реальных значений позиции БПЛА по оси X и Y соответственно. В качестве ground truth выступает информация от ультразвуковых датчиков, расположенных в помещении.

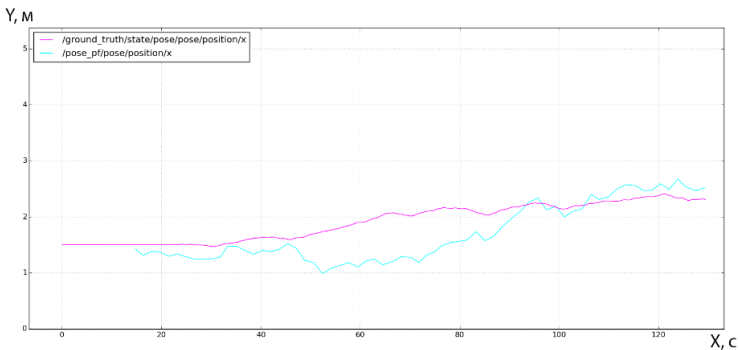


Рис. 4. Сравнение рассчитанных значений позиции квадрокоптера с реальными по оси X (красный - реальное значение, синий - рассчитанное значение)

На графике по оси Y (длинная сторона коридора) видно, что графики находятся в ощутимой близости друг к другу. Рассчитанное положение

робота практически совпадает с реальным на протяжении всего времени эксперимента.

Схожая ситуация наблюдается по оси X, но в середине эксперимента наблюдается значительное отклонение, порядка 0.4 метра. Тем не менее, на всех остальных участках отклонения не превышает ожидаемого.

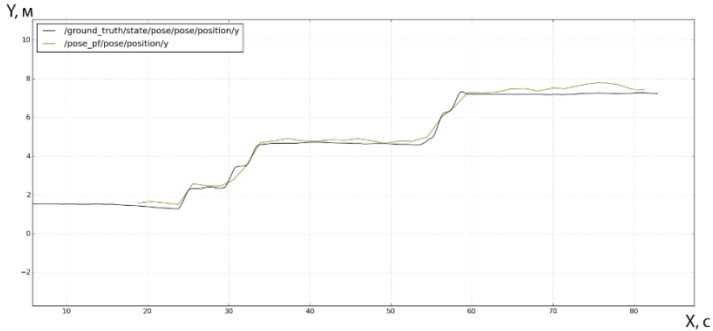


Рис. 5. Сравнение рассчитанных значений позиции квадрокоптера с реальными по оси Y (зеленый - реальное значение, черный - рассчитанное значение)

На рис. 6, 7 представлены аналогичные графики, но с использованием метода визуальной локализации, который не использует сегментацию изображения.

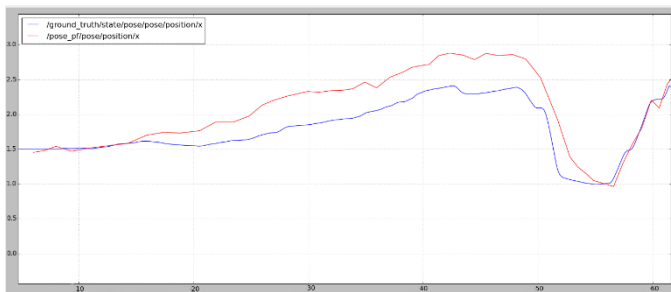


Рис. 6. Сравнение рассчитанных значений позиции квадрокоптера с реальными по оси X (без сегментации)

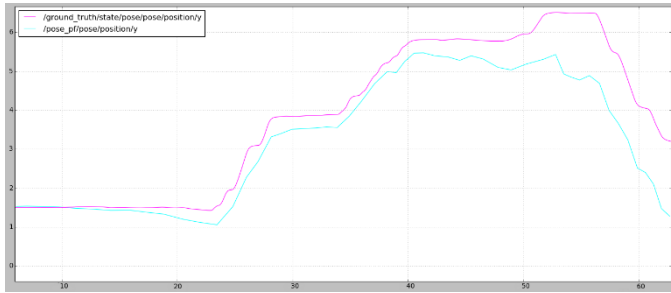


Рис. 7. Сравнение рассчитанных значений позиции квадрокоптера с реальными по оси Y (без сегментации)

## Направление дальнейших исследований

В качестве направления дальнейших исследований предполагается анализ влияния архитектуры нейронной сети на качество локализации. Предполагается создание собственной нейронной сети на базе архитектуры применяемой в нейронной сети ResNet.

## Заключение

В работе продемонстрирована состоятельность изложенного подхода для локализации. Пока рано говорить о завершенности подсистемы визуальной локализации, но на основании полученных результатов, можно сделать вывод, что использование нейросетевой сегментации позволяет повысить качество локализации по сравнению с классическими подходами. Данная задача требует дальнейшего изучения, что позволит в обозримом будущем повысить быстродействие системы и качество локализации.

## Список литературы

- [Буйвал, 2015] Alexander Buyval, Mikhail Gavrilentov, Robust indoor visual localization of unmanned micro aerial vehicle using a particle filter and 3d edge model of environment, IX международная научно-практическая «современные проблемы машиностроения» конференция СПИМ - 2015, 1–4 декабря 2015 г. Томск, Россия
- [Буйвал, 2017] Buyval A., Gavrilentov M., Magid E. A multithreaded algorithm of UAV visual localization based on a 3D model of environment: implementation with CUDA technology and CNN filtering of minor importance objects. // In Proceedings of the 2017 International Conference on Artificial Life and Robotics (ICAROB 2017). January 19-22, 2017, Miyazaki, Japan. P.356-359.

- [**Kendall, 2015**] Alex Kendall, Vijay Badrinarayanan and Roberto Cipolla "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding." arXiv preprint arXiv:1511.02680, 2015.
- [**Badrinarayanan, 2017**] Vijay Badrinarayanan, Alex Kendall and Roberto Cipolla "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation." PAMI, 2017. arXiv:1511.00561v3 [cs.CV] 10 Oct 2016
- [**Badrinarayanan, 2015**] Vijay Badrinarayanan, Ankur Handa and Roberto Cipolla "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling." arXiv preprint arXiv:1505.07293, 2015.
- [**Nuske, 2008**] Nuske, S., Roberts, J., & Wyeth, G. (2008, May). Outdoor visual localization in industrial building environments. // IEEE International Conference on Robotics and Automation, Pasadena, CA., 2008, 544-550
- [**Nuske, 2009**] Nuske, S., Roberts, J., & Wyeth, G. Robust outdoor visual localization using a three- dimensional- edge map // Journal of Field Robotics. 2009 №26(9), 728-756.
- [**Redmon, 2015**] Redmon, J., Allen, S. D. Unified, Real-Time Object Detection. arXiv preprint arXiv: 1506.02640v4, 2015.

УДК 004.855.5

## РАСПОЗНАВАНИЕ ОСНОВНЫХ ОБЪЕКТОВ ИНФРАСТРУКТУРЫ ГОРОДСКОЙ МЕСТНОСТИ ПРИ ПОМОЩИ БПЛА И НЕЙРОСЕТИ U-NET

В.А. Михайлов (*vladislav.mikhailov@phystech.edu*)  
Московский Физико-Технический Институт, Долгопрудный

О.Г. Пилипенко (*pilipenko@phystech.edu*)  
Московский Физико-Технический Институт, Долгопрудный

**Аннотация.** В статье рассматривается способ распознавания и кластеризации основных объектов жилого кампуса МФТИ посредством нейронной сверточной сети квадрокоптером. Сперва была написана нейронная сеть, задача которой распознавание уличных объектов по фотографиям, полученным с высоты 30 метров квадрокоптером. Обученная нейросеть была использована с БПЛА, который летал надо кампусом МФТИ и отслеживал изменение положений, распознанных объектов.

**Ключевые слова:** машинное обучение, компьютерное зрение, сверточная нейронная сеть, беспилотный летающий аппарат (БПЛА), Open CV, распознавание подвижных объектов.

### Введение

Машинное распознавание объектов является сложной задачей, которую не всегда можно решить привычными методами. Большое количество алгоритмов и реализованы в библиотеке Open CV. Однако для некоторых проблем, результат работы Open CV и других инструментов компьютерного зрения является неприемлемым. Так же значительная часть ныне существующих методов сложно применить на переносимых устройствах, ввиду ограниченности вычислительных мощностей.

В данной статье рассматривается БПЛА со сверточной нейронной сетью, обученной на снимках кампуса МФТИ. Нейронная сеть распознает объекты по принципу от простого к сложному:

1. Предсказывает крупномасштабное распределение объектов;
2. Таргетирует объекты и присваивает им метки;
3. Уточняет и распознает объект.

Такая многослойная структура нейросети позволяет избежать ошибки переобучения и достигнуть приемлемых результатов распознавания.

Целью данной работы является демонстрация возможностей сверточных нейросетей применительно к задачам распознавания объектов камерой БПЛА.

## 1. Описание модели эксперимента

Основными методами, используемыми в современных устройствах компьютерного зрения являются, как правило, Open CV. Зачастую их низкое качество работы, плохая обучаемость и отсутствие вычислительных мощностей у БПЛА, приводят к неудовлетворительным результатам, однако быстродействие данных алгоритмов достаточно высоко. Ввиду этих проблем, в качестве основного инструмента распознавания изображений БПЛА, была выбрана сверточная нейронная сеть с многоуровневой архитектурой, для улучшения обобщающей способности.

Так как БПЛА должен производить съемку местности на высоте от 30 до 500 метров, чтобы снимки с имеющейся камеры были в хорошем качестве. Так же аппарат должен быть устойчивым к сильным порывам ветра, которые происходят на данных высотах, чтобы повысить качество съемки и безопасность.

Ввиду причин, описанных выше создать такой БПЛА в короткие сроки не представлялось возможным, поэтому выбор арендуемого БПЛА был остановлен на Phantom 4 (рисунок 1).



Рис. 1. Квадрокоптер Phantom 4

Ниже будут приведены основные интересующие характеристики данного аппарата, а также его камеры.

Как видно из таблицы 1 все характеристики БПЛА позволяют справиться со всеми техническими проблемами. Так же данный БПЛА оснащен камерой и стабилизатором, характеристики которых приведены в таблице 2.

Таблица 1

## Характеристики Phantom 4

<b>Масса (г)</b>	<b>Высота полета (м)</b>	<b>Время полета (мин)</b>	<b>Макс. Допустимая скорость ветра (м/с)</b>	<b>Макс. Скорость (м/с)</b>
1388	До 6000	30	10	20

Таблица 2

## Характеристики камеры и стабилизатора Phantom 4

<b>Матрица</b>	<b>Угол обзора объектива в (градусах)</b>	<b>Макс размер изображ.</b>	<b>Стабилизация по 3 осям</b>	<b>Точность работы стабилизатора в (градусах)</b>
1''CMOS, 20 Мп.	84	5472 × 3078	Да	0.03

Основными процессами обработки изображения являются:

1. Захват изображения
2. Обработка изображения
3. Определение объектов и разбиение на группы

Первый пункт обработки реализуется при помощи камеры Phantom 4 и программного кода, использующего возможности библиотеки Open CV [Kevin Yu et al.]. Так мы получаем видео и при помощи данной библиотеки вырезаем из видео интересующие нас кадры таким образом, чтобы не было потеряно стыков краев каждого, вырезанного кадра. Далее происходит сама обработка изображения, попадающего на ПК по каналу Wi-Fi. Мы арендовали сервера компании Amazon, чтобы вся работа нейросети происходила в облаке. Данные передавались на сервер при помощи LTE. Так же хочется отметить, что несмотря на обработку данных в облаке, бортовой комплекс БПЛА не загружен, что позволяет организовать

высококачественное распознавание изображения. Данный подход позволяет иметь хорошие вычислительные мощности, которые требуют нейросеть u-net на небольших маломощных ПК.

И итоговым пунктом обработки изображения является обработка его нейросетью, которая была уже предварительно обучена на других похожих снимках местности. Она разбивает объекты карты на пять основных групп: люди, машины, дороги, деревья и здания. Результатом работы нейросети является карта, с четырьмя основными, распознанными классами объектов. Критерии качества работы нейросети приведены в таблице 3 в виде четырех параметров оценки каждого алгоритма.

## 2. Параметры и архитектура нейросети

Прежде чем проектировать структуру нейросети была проведена проверка существующих алгоритмов классификации на данном датасете. Ниже приведены результаты работы алгоритмов: random-forest [Anna Bosch et al. 2007], линейная регрессия, логистическая регрессия и XGboost [Chen et al., 2016]. Ниже на рисунке 2 приведены результаты распознавания каждым из этих алгоритмов примеров из созданного датасета.

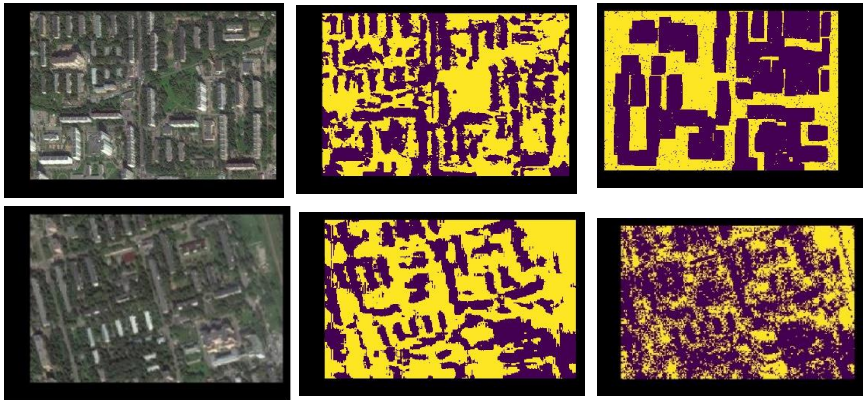


Рис. 2. Примеры обучения нейросети распознаванию строений известными алгоритмами (слева – исходные фото, по центру – линейная регрессия, справа – random-forest)

Из рисунка 2 видно, что ни один алгоритм не может предоставить качественные результаты. Так в случае использования линейной регрессии строения распознаются неточно с границами меньше чем это есть на картинке с БПЛА. В то же время реализация алгоритма random-forest определяет каждый дом больше, чем он есть, что приводит к слипанию



домов (рис. 2 справа). На кросс-валидации Random-forest показывает сильное зашумление.

В таблице три изложены основные параметры, которые были получены при сравнении параметров работы известных алгоритмов, с проектируемой сетью u-net, в применении к данной задаче.

Таблица 3

Численные характеристики параметров, полученные в результате экспериментов

Название алгоритма	Train accuracy	Validation accuracy	Скорость обучения(с)	Скорость предсказания(мс)
Random-forest	0,994	0,583	2,98	247
Linear reg.	0,787	0,649	1,344	68
XGboost	0,973	0,785	12,75	137
U-Net	0,989	0,954	>> 1 мин.	322

Как можно видеть из таблицы 3, по подавляющему большинству параметров лучшим, для решения этой задачи, является алгоритм U-Net. Эти результаты предопределили наш выбор в пользу данной архитектуры.

## 2.1 Архитектура нейросети

Главной задачей, проектируемой сверточной нейросети является распознавание четырех основных классов объектов: строения, дороги, зеленые массивы, вода.

Основой архитектуры, данной нейросети является нейросеть u-net [Chen et al., 2016]. На рисунке 3 приведена блок-схема нейросети.

Для предсказания шаблона- маски использовалось обычное скользящее окно 3 на 3. Каждая из картинок нарезалась на патчи, которые предсказываются и собираются в исходную. Итогом работы данного этапа предсказания являются вероятности того, что в конкретном пикселе находится каждый из шести классов. Чтобы получить бинарную маску обучения мы вводим фильтр с уровнем пропуска 0,42, который был подобран экспериментально.

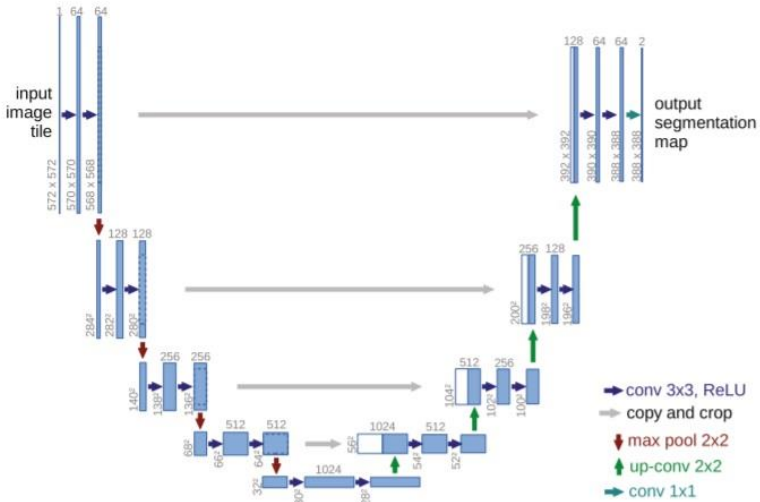


Рис. 3. Схема, реализуемой нейросети u-net

Архитектура сети показана на рисунке 3. Она состоит из контрактного пути (левая сторона) и расширенного пути (правая сторона). Контрактный путь является типичной архитектурой сверточной сети. Он состоит из повторяющихся применений двух сверток  $3 \times 3$  (неуплотненные свертки), с активатором ReLU. Далее производится операция объединения  $2 \times 2$  с максимальным шагом 2, для понижающей дискретизации. На каждом понижающем шаге дискретизации мы удваиваем число признаков в каналах. Каждый шаг расширенного пути состоит из повышающей дискретизации, после чего следует свертка  $2 \times 2$  («up-convolution»), которая уменьшает вдвое количество каналов функции, далее следует склейка с соответствующим обрезанным образом карты признаков из левой части сети и две свертки  $3 \times 3$ , с активатором ReLU. Обрезка необходима из-за потери граничных пикселей в каждой свертке. На конечном слое свертка  $1 \times 1$  используется для отображения каждого 64 - компонентного вектора признаков в необходимое количество классов. Всего сеть имеет 23 сверточных слоя. Чтобы обеспечить бесшовную разбивку карты сегментации вывода (см. Рисунок 2), важно выбрать размер входного фрагмента так, чтобы все операции с максимальным пулом  $2 \times 2$  применялись к слою с четным размером по x и y.

## 2.2. Обучение нейросети

Для обучения нейросети было использовано 20 снимков территории города Долгопрудный, которые были размечены на 4 основных распознаваемых классов (см. п. 2.1) вручную. Каждое изображение было отмасштабировано к одному размеру 900 на 900 пикселей.

Входные изображения и соответствующие им карты сегментирования используются для обучения сети с реализацией стохастического градиентного спуска Theano [Yichuan Tang, 2013]. Из-за особенностей сверток, выходное изображение меньше входного на постоянную ширину границы. Чтобы минимизировать накладные расходы и максимально использовать памяти GPU, мы отдаем предпочтение большим входным фрагментам за большой размер партии и, следовательно, уменьшите партию до одного изображения. Соответственно, мы используем большой импульс (0,99) так, чтобы большое количество ранее рассмотренных учебных образцов определяло обновление в текущем шаге оптимизации.

Для оценки функции потерь проектируемой нейросети используется логистическая функция потерь (log loss multiclass). Подсчет ошибки производится по формуле:

$$F_{loss} = -\frac{1}{N} \sum_i^N \sum_j^M y_{ij} \ln(p_{ij})$$

где  $N$  – количество экземпляров,  $M$  – количество разных меток,  $y_{ij}$  – двоичная переменная с ожидаемыми метками,  $p_{ij}$  – вероятность классификации, выдаваемая классификатором для  $i$ -го экземпляра и  $j$ -ой метки.

Мы предварительно вычисляем карту весов для каждой основной сегментации, чтобы компенсировать различную частоту пикселей от определенного класса в наборе данных обучаемых и принудить сеть к изучению малых классов.

Граница раздела вычисляется с использованием морфологических операций. Затем карта весов вычисляется по формуле:

$$w(x) = w_c(x) + w_0 \cdot \exp\left(-\frac{(d_1(x) + d_2(x))^2}{2\sigma^2}\right)$$

где  $w_c(x)$  – карта весов для компенсирования частот классов,  $w_0 = 5$  в нашем эксперименте,  $\sigma=5$  пикселей,  $d_1(x)$  - расстояние до границы ближайшей ячейки,  $d_2(x)$  – расстояние до границы второй ближайшей ячейки.

В глубоких сетях со многими сверточными слоями и разными путями в сети, очень важна инициализация весов. В противном случае, части сети могут иметь чрезмерную активацию, в то время как другие части никогда не вносят свой вклад. В идеале первоначальные веса должны быть

адаптированы таким образом, чтобы каждая карта функции в сети имела единичную дисперсию. Для сети с нашей архитектурой (чередующиеся свертки и слои ReLU), это может быть достигнуто путем отрисовки начальных весов из гауссовского распределения со стандартным отклонением  $\sqrt{2/N}$ , где N обозначает количество входящих узлов одного нейрона[5]. Например, для свертки 3x3 и 64 функциональных каналов на предыдущем уровне  $N = 9 \cdot 64 = 576$ .

Лучшие показатели нейросети U-Net во время тренировки были при реализации 64 эпох обучения (при данном количестве появлялся нормальный результат):

- Качество тренировки - 0,9889
- Train loss - 0,0104
- Качество на валидации 0,954
- Validation loss 0,117

### 3. Примеры распознавания и сегментации объектов инфраструктуры

На рисунках 4 - 7 приведены результаты работы сверточной нейросети. Каждый результат представляет собой три картинки, снимок с камеры, распознавание человеком и распознавание какого-либо класса компьютером. В результате получили качественные предсказания для четырех классов из пяти.

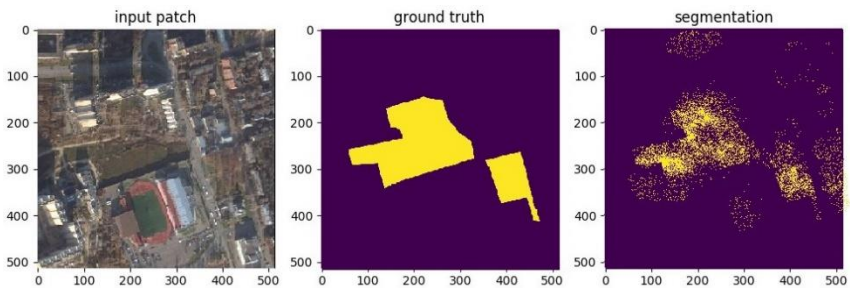


Рис. 4. Распознавание зеленых массивов

Одними из самых неточных результатов получился при распознавании зеленых массивов (см. рис.4). Видно, что сеть способна вычленять основные скопления деревьев, а также давать положения более мелких скоплений. Однако следует отметить большую зашумленность результата случайными точками, из-за неточности разметки и большого разброса деревьев.

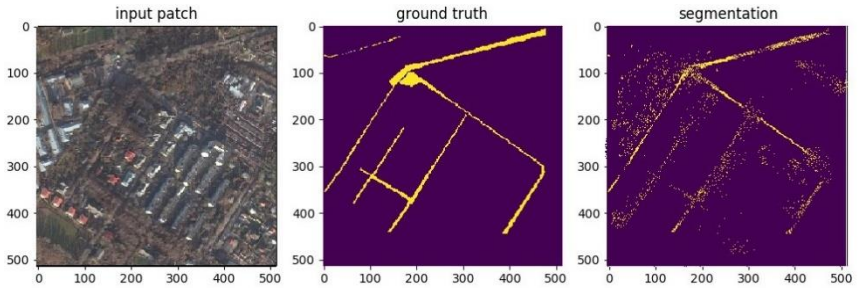


Рис. 5. Распознавание дорог

Вторым классом по качеству распознавания мы отнесли класс дорог (см. рис.5). Здесь присутствует большая зашумленность и периодически непрерывность дорог пропадает, мы считаем данный результат нормальным, данные результаты связываем с тем, что градиентные фильтры, которые присутствуют в нейросети не могут четко определять перепад между жилой застройкой и проезжей частью.

Самые лучшие результаты были продемонстрированы при распознавании жилых построек (см рис. 6-7). Спроектированная нейросеть смогла находить некоторые дома, которые не получалось найти изначально человеческим глазом. Несмотря на большой уровень зашумления основные дома были распознаны, за исключением центрального дома в левом верхнем углу рисунка 6. Связываем данный результат с тем, что в обучающей выборке был всего лишь один дом с оранжевым цветом, что не достаточно для нейросети, чтобы обучиться.

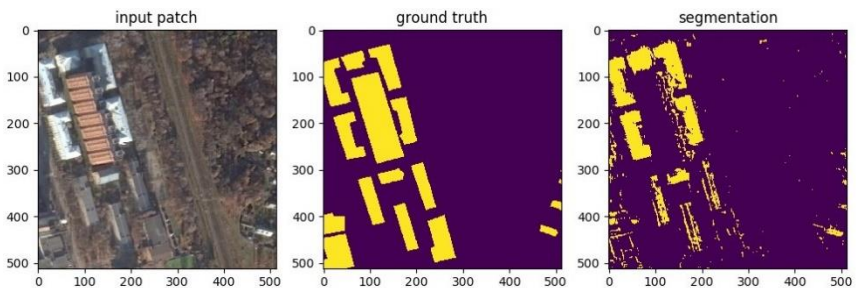


Рис. 6. Распознавание домов

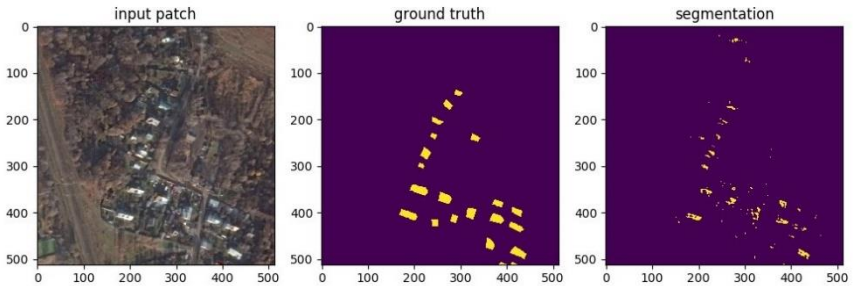


Рис. 7. Распознавание домов

Наилучшие результаты были достигнуты для класса строений, а наихудшие для класса дорог. Так, например, обученная нейросеть, распознавала все имеющиеся на фотографии дома, однако не всегда точно удавалось определять границы строений. Как можно видно из рисунка данный тип фильтрации нейросети не позволяет полностью избавиться от шумов, размеры которых могут выражаться в виде точек в 2-3 пикселя, что очень сильно затрудняет распознавание класса автомобилей.

#### 4. Достигнутые результаты

Основным достижением авторы считают создание системы, которая в режиме реального времени способна распознавать основные элементы инфраструктуры местности (строения, зеленые массивы, дороги, вода). Основной принцип работы данной системы изложен в пункте 1 данной статьи.

Как было показано в пункте 2 данной статьи, нейросеть U-Net обладает наилучшим качеством распознавания, применительно к данной задаче. Поэтому авторы считают, что одной из возможностей для дальнейшего развития – расширение возможностей данной системы для распознавания объектов на различных типах местности.

После анализа полученных результатов распознавания (см. пункт 3) были сделаны выводы, позволяющие улучшить качество данного исследования. Условия улучшения результатов:

- увеличение качества изображения;
- увеличение размера обучающей выборки;
- улучшение качества разметки, увеличение числа слоев и фильтров в сверточных слоях, для чего необходимо более мощное оборудование;
- добавление дроп-аута (Dropout).

К сожалению, на данный момент, приведенное, решение вопроса распознавания объектов при помощи БПЛА, невозможно встроить в программно-аппаратный комплекс малых БПЛА, однако при использовании мощных вычислительных машин на земле или облачных технологий, данное решение имеет возможность для реализации.

При выполнении вышеперечисленных условий, применение данного вида нейросети возможно в картографических, розыскных мероприятиях,

**Благодарности.** Авторы считают своим приятным долгом поблагодарить д.ф.-м.н., профессора В. Е. Павловского, который направлял, консультировал и помогал данному исследованию.

## Список литературы

- [**O Ronneberger et al., 2015**] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation.
- [**Anna Bosch et al. 2007**] A. Bosch, A. Zisserman, X. Munoz. Image Classification using Random Forests and Ferns, 2007.
- [**Chen et al., 2016**] Tianqi Chen, Carlos Guestrin., XGBoost: A Scalable Tree Boosting System
- [**Kevin Yu et al.**] Kevin Yu Rahman Kadierring Umesh Dinkar., XGBoost: Capturing Webcam Images to Display on a Web Interface
- [**Yichuan Tang, 2013**] Yichuan Tang, XGBoost: Deep Learning using Linear Support Vector Machines
- [**Martin Längkvist et al., 2016**] Martin Längkvist, Andrey Kiselev, Marjan Alirezaie and Amy Loutfi: Classification and Segmentation of Satellite Orthoimagery Using Convolutional Neural Networks
- [**Idris Kahraman et al., 2015**] Idris Kahraman, Muhammed Kamil Turan, and Ismail Rakip Karas: Road Detection from High Satellite Images Using Neural Networks <http://lasagne.readthedocs.io/en/latest/index.html> - documentation for Lasagne – good framework for neural network

УДК 004.932.7, 004.4, 004.021

## МЕТОД РЕШЕНИЯ ЗАДАЧИ ОДНОВРЕМЕННОГО КАРТИРОВАНИЯ И ЛОКАЛИЗАЦИИ ПО ВИДЕОПОТОКУ И ДАННЫМ ИНЕРЦИАЛЬНОЙ НАВИГАЦИОННОЙ СИСТЕМЫ ДЛЯ МАЛЫХ БПЛА

А.В. Боковой (*bokovoy@isa.ru*)

Федеральный исследовательский центр “Информатика и управление” Российской академии наук, Москва

**Аннотация.** В статье рассматривается метод решения задачи одновременного картирования и локализации малого беспилотного летательного аппарата по данным, полученным с единственной камеры и датчиков инерциальной навигационной системы. Приведено экспериментальное исследование разработанного алгоритма и проведен анализ полученных результатов на актуальных коллекциях данных.<sup>1</sup>

**Ключевые слова:** SLAM, ИНС, методы одновременного картирования и локализации, видеопоток, единственная камера.

### Введение

В последнее время, для успешной навигации малых беспилотных летательных аппаратов активно применяются методы одновременного картирования и локализации (vision-based Simultaneous Localization and Mapping – vSLAM) по видеопотоку [Blösch M. et al, 2010][Fraundorfer F. et al, 2012][Yang S et al, 2016]. Подобный выбор обуславливается как ограниченным количеством датчиков, которые можно установить на малые беспилотные летательные аппараты, так и условиями, в которых происходит навигация (например, в помещениях датчики GPS/ГЛОНАСС не работают). С развитием электроники стало возможным производить нетрудоёмкие вычисления на борту малых БПЛА, но этого по-прежнему недостаточно для полноценной обработки алгоритмов компьютерного зрения, к которым относятся методы vSLAM, в реальном времени. Тем не менее, мощности современных бортовых вычислителей вполне хватает для сжатия видеопотока, полученного с единственной камеры, установленной

---

<sup>1</sup> Работа выполнена при финансовой поддержке РФФИ (проект № 15-07-07483)



на БПЛА, с последующей передачей его на рабочую станцию по беспроводному каналу и дальнейшей обработкой алгоритмами vSLAM.

Для повышения степени автономности малых БПЛА необходимо увеличить один из ключевых параметров алгоритмов vSLAM, а именно – быстродействие, при сохранении точности построения пройденной траектории с картой и локализации. Также, используя методы одновременного картирования и локализации по видеопотоку, следует учитывать накапливающиеся по мере работы алгоритма ошибки, которые возникают в силу использования вероятностных моделей на этапах обработки видеопотока, при составлении карты и локализации, а также из-за не совершенности камер (часто присутствуют помехи, искажения и т.д.). Для коррекции таких ошибок существует семейство алгоритмов, называемых алгоритмами замыкания траекторий (loop-closure algorithms)[Angeli A. et al, 2008][Whelan T. et al, 2013]. Такие алгоритмы позволяют скорректировать построенную карту и пройденную траекторию при условии того, что робототехническое устройство (в конкретном случае – малый беспилотный летательный аппарат) проходит через уже известное место (т.е. место, для которого уже построена локальная карта и траектория), а также успешно его распознает.

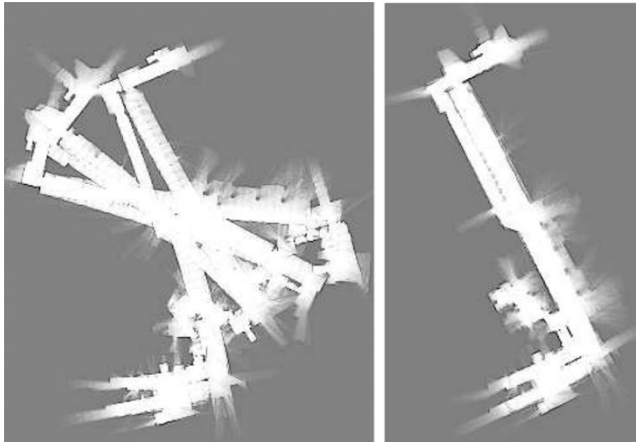


Рис. 1. Пример результата работы алгоритма loop-closure для задачи SLAM

Основным недостатком алгоритмов loop-closure является их недостаточное для функционирования в реальном времени быстродействие. Наиболее затратной частью алгоритма является перестроения карты и нахождение уже пройденных мест. Один из выходов из сложившейся ситуации – запуск алгоритмов vSLAM и loop-closure параллельно при понижении частоты loop-closure до 1-2 Гц. Даже при

предложенном подходе, современные бортовые вычислители для малых БПЛА не способны обрабатывать всю информацию в реальном времени.

Одним из выходов из сложившейся ситуации является дополнение информации с камеры малого беспилотного летательного аппарата данными с инерциальной навигационной системы (ИНС) и полное или частичное исключение алгоритма loop-closure из метода vSLAM. В выбранном для дальнейших исследований [Боковой А.В, 2015] методе LSD-SLAM [Engel J. et al, 2014] на этапе составления карты и реконструкции пройденной траектории используется расширенный фильтр Калмана (extended Kalman filter – EKF)[Ljung L, 1979], где управляющий вектор заранее не известен и вычисляется посредством алгоритмов компьютерного зрения. При использовании данных с ИНС отпадает надобность в вычислении управляющего вектора, т.к. его параметры можно получить из данных акселерометра (после устранения шумов)[El-Sheimy et al, 2004] при меньших вычислительных затратах. Еще одним подходом может быть корректировка управляющего вектора, рассчитанного методом vSLAM, с помощью данных ИНС, т.е. вычисление управляющего вектора на каждой итерации алгоритма происходит параллельно как для видеопотока, так и для ИНС. Такой подход позволит увеличить точность построения пройденной траектории и карты окружающей среды, но увеличит время работы алгоритма, что может сделать применение такого алгоритма на бортовых вычислителях малых беспилотных летательных аппаратов – невозможной.

Также, отрицательной особенностью работы многих алгоритмов vSLAM является затруднение локализации и построения карты при потере связи текущего кадра с предыдущими [Williams B. et al, 2007], поскольку процесс работы полностью завязан на сравнении текущего кадра с уже имеющимися. При потере связи между кадрами, требуются дополнительные алгоритмы для восстановления позиции робототехнической системы, ее релокации и перестроения карты. Поставленную проблему также можно решить с помощью данных, полученных с ИНС, поскольку нам всегда будет известна ориентация камеры (данные с гироскопа) относительно предыдущего положения.

## **1. Краткое описание метода**

Основная идея для разработанного метода одновременного картирования и локализации по данным видеокамеры и инерциальной навигационной системы состоит в замене или дополнении расчетов расширенного фильтра Калмана на этапах предсказания и коррекции данными с датчиков ИНС. Для этого, на этапе локализации и составления карты, в алгоритм LSD-SLAM были добавлены следующие сценарии:

- 1) Частичная замена управляющего вектора данными, рассчитанными с помощью датчиков инерциальной навигационной системы, при потере связи алгоритма с предыдущей точкой наблюдения.
- 2) Усреднение управляющих векторов, полученных с помощью камеры и с помощью ИНС.
- 3) Полная замена управляющего вектора данными, рассчитанными с помощью датчиков инерциальной навигационной системы, на всех этапах работы алгоритма.

Метод расчета управляющего вектора для расширенного фильтра Калмана подробно описан в работе [Nützi G. et al, 2011]. Для реализации метода IMU+LSD-SLAM, предложенный метод был доработан с учетом возможности его практической реализации и работы с моделируемыми данными.

## **2. Реализация алгоритма**

Алгоритм был реализован под операционную систему Linux на языке C++. В качестве рабочей среды был использован популярный для решения робототехнических задач фреймворк Robot Operating System (ROS). В качестве основы была взята открытая реализация метода LSD-SLAM [[https://github.com/tum-vision/lst\\_slam](https://github.com/tum-vision/lst_slam)].

## **3. Экспериментальное исследование**

Эксперимент проводился в среде Gazebo с использованием фреймворка ROS. В качестве модели среды для проведения испытаний была выбрана модель помещения, созданная для проведения конкурса RoboCup Rescue Simulation League 2013, которая была адаптирована для Gazebo.

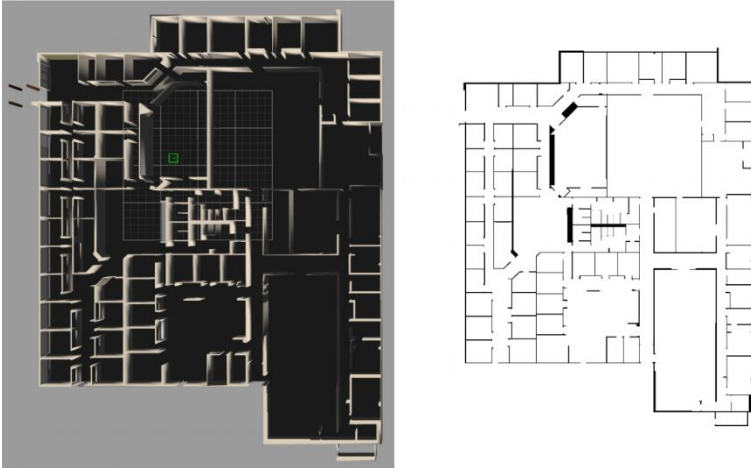


Рис. 2. Модель среды, использовавшаяся для экспериментального исследования метода. Слева – вид модели сверху. Справа – план модели помещения

Помещение состоит из набора комнат и коридоров разной площади. Для упрощения работы с моделью, мебель и дополнительные препятствия отсутствуют. Общая площадь – 6 300 м<sup>2</sup> (задана вручную).

В качестве робототехнического устройства, которое будет производить картирование и локализацию в представленной среде, был выбран стандартный малый беспилотный летательный аппарат hector\_quadrotor.

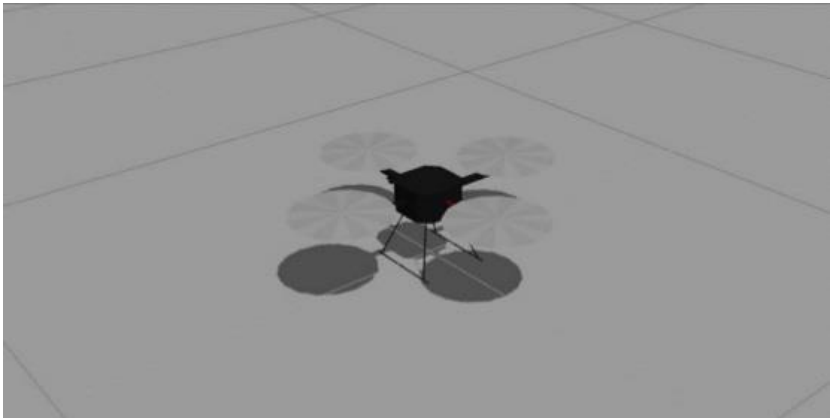


Рис. 3. Модель малого беспилотного летательного аппарата hector\_quadrotor в среде Gazebo

На hector\_quadrotor была смоделирована работа следующих датчиков:

- 3-х осевой гироскоп;
- 3-х осевой акселерометр;
- магнетометр;
- монокулярная камера с заданной матрицей внутренней калибровки.

Частота обновления данных для инерциальной навигационной системы составляет 200 Гц, для монокулярной камеры – 30 Гц. Разрешение изображений, полученных с камеры составляет 640x480 пикселей. Динамическая модель робота задается стандартными значениями, используемыми для hector\_quadrotor [Meyer J. et al, 2012].

В качестве сценариев для тестирования метода были составлены 10 траекторий на выбранной модели местности. Длина траекторий варьируется от 25 м до 183 м. На выборках присутствуют как траектории с замыканиями (7 шт.), так и без (3 шт.). Предполагается, что беспилотный летательный аппарат движется по заданным траекториям равномерно со скоростью 1 м/с, а угловая скорость рысканья не превышает 0,5 об/с. Данные для датчиков ИНС и камеры были составлены при “ручном” полете по каждой из траекторий при заданных частотах обновлений датчиков. Зашумленность данных моделируется за счет динамической системы модели малого беспилотного летательного аппарата, которая учитывает естественные колебания при полете.

Результат работы разработанного метода сравнивается (IMU+LSD-SLAM) с методом LSD-SLAM. Основными параметрами для сравнения являются:

- скорость работы алгоритма;
- точность построенной траектории;
- точность построенной карты.

В качестве параметра для измерения точности построенной траектории было выбрано среднее отклонение от эталонной (ground-truth) траектории. Эта величина является средним разности расстояния между построенной методом в каждый момент времени траектории и реально пройденной беспилотным летательным аппаратом траектории в метрах.

Для тестирования была выбрана реализация, в которой данные ИНС полностью заменяют расчет управляющего вектора в расширенном фильтре Калмана. В таблице 6.1 представлены результаты экспериментального исследования.

Для большей наглядности можно представить полученные данные в виде графиков.

Результаты тестирования разработанного метода и LSD-SLAM

Сценарий	Длина (м)	Замыкания	Среднее время обработки кадра (сек)		Среднее отклонение от эталонной траектории (м)		Среднее отклонение от эталонной карты (м)	
			LSD-SLAM	IMU+LSD-SLAM	LSD-SLAM	IMU+LSD-SLAM	LSD-SLAM	IMU+LSD-SLAM
<b>traj_1</b>	25,3	-	0,189	0,137	0,23	0,24	0,35	0,34
<b>traj_2</b>	28,9	+	0,211	0,141	0,18	0,13	0,52	0,5
<b>traj_3</b>	39,2	+	0,231	0,156	0,33	0,34	0,57	0,51
<b>traj_4</b>	58,3	+	0,288	0,135	0,38	0,34	0,56	0,53
<b>traj_5</b>	59,2	+	0,29	0,143	0,41	0,37	0,62	0,57
<b>traj_6</b>	91,5	-	0,192	0,149	0,81	0,6	0,93	0,71
<b>traj_7</b>	97,2	+	0,291	0,147	0,69	0,63	0,85	0,77
<b>traj_8</b>	101	+	0,295	0,143	0,71	0,65	0,86	0,78
<b>traj_9</b>	110,2	-	0,185	0,159	0,73	0,66	1,12	0,89
<b>traj_10</b>	125,3	+	0,315	0,141	0,78	0,69	0,98	0,91

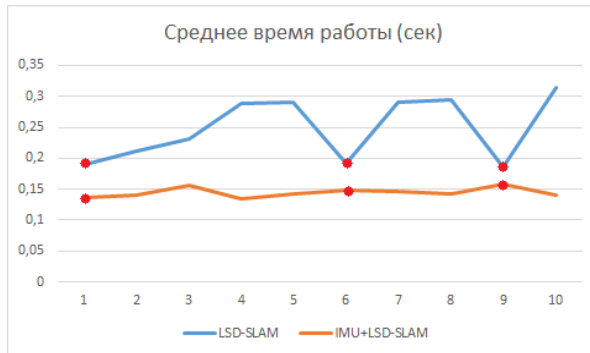


Рис. 4. Среднее время работы LSD-SLAM и IMU+LSD-SLAM. Ось X – номер сценария, ось Y – время работы в сек. Красным выделены скорости работы на незамкнутых траекториях

Как видно из полученного графика, время работы алгоритма IMU+SLAM стабильно увеличивается с каждой коллекцией. Такое поведение связано с тем, что длина траектории увеличивается от коллекции к коллекции и время, необходимое на обработку накопленных данных также увеличивается. В то время как время работы алгоритма LSD-SLAM сильно зависит от наличия замыкания траектории, поскольку в таком случае к работе алгоритма подключается метод замыкания траекторий (loop-closure algorithm), который корректирует построенную карту и пройденную траекторию, при этом затрачивая на корректировку ощутимое

количество времени. В сценариях без замыканий (сценарии traj\_1, traj\_6, экфо\_9), скорость работы LSD-SLAM заметно выше, чем с замыканиями.

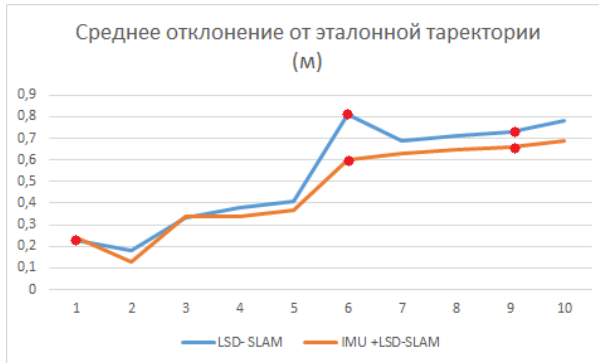


Рис. 5. Среднее отклонение траекторий, построенных алгоритмами LSD-SLAM и IMU+LSD-SLAM от эталонных. Красным выделены отклонения на незамкнутых траекториях

Среднее отклонение полученной траектории от эталонной для обоих методов равномерно возрастает в зависимости от длины пройденной траектории. Существенная разница видна лишь на коллекции traj\_6. Резкое повышение отклонения в случае с traj\_6 для метода LSD-SLAM можно объяснить тем, что в сценарии присутствуют резкие повороты и алгоритм корректировки траектории тратит много времени на восстановление пути и карты.

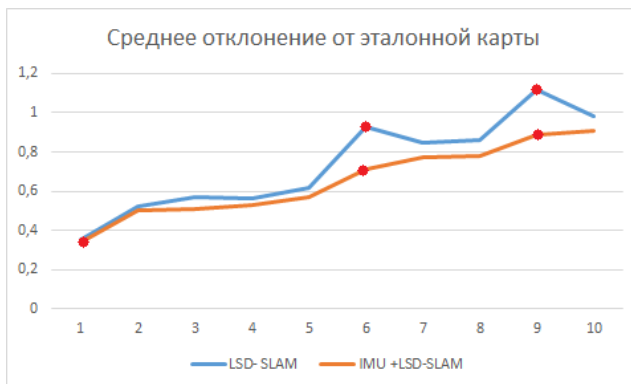


Рис. 6. Среднее отклонение построенной карты от эталонной для методов LSD-SLAM и IMU+LSD-SLAM. Красным выделены отклонения на незамкнутых траекториях

Среднее отклонение построенной методами LSD-SLAM и IMU-SLAM карты от эталонной также возрастает в зависимости от длины траектории. Скачки в сторону увеличения отклонения для метода LSD-SLAM наблюдаются на сценариях без замыканий траектории, поскольку корректировки карты в таком случае не происходит и погрешность вычислений продолжает накапливаться. Метод IMU+LSD-SLAM в таких условиях работает более стабильно.

## Заключение

В результате работы был разработан метод, основанный на алгоритме LSD-SLAM, учитывающий информацию, полученную с датчиков инерциальной навигационной системы малого беспилотного летательного аппарата. Работоспособность алгоритма была протестирована на специально разработанных модельных данных. Экспериментальное исследование показало, что предложенный метод позволил получить прирост производительности в среднем на 30% (44,5% в лучшем случае) по сравнению с LSD-SLAM, при этом повысив точность построения траектории на 10% и карты на 7%. Таким образом, можно заключить, что предложенный метод может быть успешно использован в задачах навигации малых беспилотных летательных аппаратов, оснащенных единственной камерой и датчиками инерциальной навигационной системы, в том числе с возможностью производить вычисления на борту, что позволит повысить степень автономности таких робототехнических устройств.

## Список литературы

- [Blösch M. et al. 2010] Blösch, M., Weiss, S., Scaramuzza, D., & Siegwart, R. Vision based MAV navigation in unknown and unstructured environments //Robotics and automation (ICRA), 2010 IEEE international conference on. – IEEE, 2010. – С. 21-28
- [Fraundorfer F. et al., 2012] Fraundorfer, F., Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P., & Pollefeys, M. Vision-based autonomous mapping and exploration using a quadrotor MAV //2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. – IEEE, 2012. – С. 4557-4564.
- [Yang S. et al, 2016] Yang S., Scherer S. A., Zell A. Robust Onboard Visual SLAM for Autonomous MAVs //Intelligent Autonomous Systems 13. – Springer International Publishing, 2016. – С. 361-373



- [**Angeli A. et al, 2008**] Angeli, A., Filliat, D., Doncieux, S., & Meyer, J. A. Fast and incremental method for loop-closure detection using bags of visual words //IEEE Transactions on Robotics. – 2008. – Т. 24. – №. 5. – С. 1027-1037
- [**Whelan T. et al, 2013**] Whelan, T., Kaess, M., Leonard, J. J., & McDonald, J. Deformation-based loop closure for large scale dense RGB-D SLAM // Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. – IEEE, 2013. – С. 548-555.
- [**Боковой А.В, 2015**] Боковой А.В. Исследование методов одновременного картирования и локализации беспилотных летательных аппаратов по видеопотоку, полученному с единственной камеры // Второй Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2015, 9 октября 2015г., г. Санкт-Петербург, Россия): Труды семинара. – Санкт-Петербург: Изд-во «Политехника-сервис», 2015. С. 26-33.]
- [**Engel J, et al, 2014**] Engel J., Schöps T., Cremers D. LSD-SLAM: Large-scale direct monocular SLAM //European Conference on Computer Vision. – Springer International Publishing, 2014. – С. 834-849.]
- [**Ljung L, 1979**] Ljung L. Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems //IEEE Transactions on Automatic Control. – 1979. – Т. 24. – №. 1. – С. 36-50.
- [**El-Sheimy et al, 2004**] El-Sheimy N., Nassar S., Noureldin A. Wavelet de-noising for IMU alignment //IEEE Aerospace and Electronic Systems Magazine. – 2004. – Т. 19. – №. 10. – С. 32-39.
- [**Williams B., et al, 2007**] Williams B., Klein G., Reid I. Real-time SLAM relocation //2007 IEEE 11th International Conference on Computer Vision. – IEEE, 2007. – С. 1-8
- [**Nützi G. et al, 2011**] Nützi, G., Weiss, S., Scaramuzza, D., & Siegwart, R. Fusion of IMU and vision for absolute scale estimation in monocular SLAM //Journal of intelligent & robotic systems. – 2011. – Т. 61. – №. 1-4. – С. 287-299.
- [**Meyer J. et al. 2012**] Meyer, J., Sendobry, A., Kohlbrecher, S., Klingauf, U., & Von Stryk, O. Comprehensive simulation of quadrotor uavs using ros and gazebo //International Conference on Simulation, Modeling, and Programming for Autonomous Robots. – Springer Berlin Heidelberg, 2012. – С. 400-411.

УДК 528.854

## СИСТЕМА КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ РАСПОЗНАВАНИЯ ЭЛЕМЕНТОВ УПРАВЛЕНИЯ АВТОМОБИЛЕМ РОБОТОМ-ВОДИТЕЛЕМ

П.С. Сорокоумов (*petr.sorokoumov@gmail.com*)  
НИЦ "Курчатовский институт", Москва

**Аннотация.** В данной работе описывается метод распознавания рулевого колеса и рычага переключения передач автомобиля, предназначенный для использования антропоморфным роботом-водителем. Разработанный алгоритм позволяет корректно обрабатывать значительные перепады яркости изображения и не требует изменения конструкции автомобиля. Полученный программный комплекс в первую очередь предназначен для разработки биоподобных методов управления движениями антропоморфных роботов, но может быть в дальнейшем использован как самостоятельный компонент систем компьютерного зрения.

**Ключевые слова:** компьютерное зрение, биоподобие, робот-водитель

### Введение

Применение антропоморфных роботов на практике существенно осложнено тем, что многие привычные для человека задачи, например координация сложных движений, решаются ими недостаточно качественно. Для проведения исследований по управлению движением конечностей на примере действий водителя оказалось необходимым реализовать вспомогательную систему компьютерного зрения, способную определять по данным с видеокамеры робота органы управления автомобилем и рассчитывать их положение относительно робота. Полученная система может быть использована для распознавания органов управления этих или схожих типов в других задачах, требующих автоматизации производственных процессов.

### 1. Постановка задачи

Биоподобие – один из главных источников новых идей в области робототехники; например, сами понятия об интеллекте, интеллектуальном агенте возникли во многом за счёт обобщения характерных свойств

биологических систем. Среди других заимствованных из биологии концепций – построение роботов, строение тело которых аналогично животным, в том числе человеку. Однако системы управления движениями этих роботов не всегда решают свою задачу качественно [Pfeifer, 2012]; одними из самых сложных задач были и остаются те, которые взрослый человек выполняет без труда – движение по пересечённой местности, посадка в автомобиль, манипуляции предметами обстановки ([Ackerman,2015], [Atkeson, 2015]).

Для того чтобы улучшить работу систем управления роботами в таких условиях, желательно поставить перед андроидом просто сформулированную, но плохо решаемую на текущий момент задачу. Например, можно попытаться воспроизвести работу человека-водителя с органами управления автомобилем.

Такая постановка задачи имеет несколько преимуществ перед, например, движениями по пересечённой местности или по лестнице: робот находится в сидячем положении, поэтому поддерживать равновесие ему не нужно; робот движется в ограниченном пространстве, которое легко наблюдается; возможно сравнение движения рук робота и человека-водителя при выполнении схожих задач. Для исследований в этой области в лаборатории робототехники НИЦ "Курчатовский институт" был создан экспериментальный стенд, показанный на рис. 1. Из органов управления на стенде имеются рулевое колесо, рычаг переключения передач и педаль; все элементы управления сделаны легкоподвижными, чтобы снизить риск повреждений робота.



Рис. 1. Общий вид робота Reem-C [Reem-C] за экспериментальным стендом

Для того чтобы робот смог манипулировать органами управления автомобилем, он должен сначала определить их положение в пространстве. Разумно будет построить для этого систему компьютерного зрения; при этом полученное решение можно будет применять не только для лабораторных исследований на конкретном роботе, но и для более широкого круга задач. Крайне желательно не вносить изменений в конструкцию органов управления, потому что иначе система может снизить надёжность органов управления и потребует для развёртывания в новых условиях, например, в реальном автомобиле, слишком больших усилий. Готовых универсальных решений для этой задачи распознавания в настоящий момент нет.

Далее в данной работе описаны основные решения, применённые при создании системы распознавания органов управления. По результатам обзора готовых средств и методов, решающих аналогичные проблемы, были определены наиболее перспективные пути построения системы в условиях данной задачи. Далее были построены описания рулевого колеса и рычага, достаточные для их успешной локализации. Разработанные модели были реализованы в виде модулей системы ROS и протестированы на видеоданных, полученных от камер тестового робота. Полученное решение оказалось пригодным для использования в проводимых в настоящий момент экспериментах.

## **2. Выбор компонентов алгоритма распознавания**

Автоматическое распознавание элементов управления автомобилем – задача весьма трудная. Прежде всего, внешний вид и геометрия руля и рычага переключения передач сильно отличается в разных моделях; зачастую водители меняют их вид ещё и самостоятельно, используя чехлы, оплётки и т.п. Форма рулевого колеса, кроме того, может быть не круглой, но на практике подобное встречается достаточно редко и главным образом в специализированных спортивных автомобилях, так что в дальнейшем такая возможность не рассматривается. Салон автомобиля, то есть фон, на котором располагаются органы управления, может ещё больше отличаться от модели к модели, и модифицируется он чаще. Это означает, что система распознавания должна учитывать особенности конкретно используемой машины при составлении модели элемента управления.

Ещё один важный фактор сложности задачи – то, что внешний вид органов управления зависит от окружающей обстановки, прежде всего от освещения. При движении яркость освещения часто крайне нестабильна, полосы тени и света от сменяют друг друга с высокой скоростью. При этом многие значимые источники света движутся относительно автомобиля с большой скоростью (фары встречных машин, фонари). В ночных условиях,

особенно при выключенном освещении салона, даже человеку сложно распознать рулевое колесо и рычаг переключения передач визуально. Поэтому в разрабатываемом распознавателе необходимо использовать как один из признаков некоторую величину, которая слабо меняется при колебаниях освещённости. Крайне желательно также, чтобы использованный метод допускал работу в тёмное время суток.

Внешний вид руля и рычага меняется также в зависимости от положения рук водителя на них. Водитель-человек при этом способен благодаря осязанию чувствовать контакт с элементами управления, то же самое может и водитель-робот при наличии тактильных сенсоров.

Наконец, значительные конструктивные переделки тестового автомобиля крайне нежелательны. Разрабатываемое решение должно работать с внесением только минимальных быстрых обратимых изменений в салон; глубокое переоборудование не соответствует постановке задачи.

Исходя из изложенных особенностей задачи, можно обосновать выбор алгоритма для её решения.

- можно применить геометрические параметры салона для поиска органов управления на видеопотоке. Но, во-первых, положение робота при работе может изменяться от тряски, и проконтролировать эти перемещения сложно; во-вторых, достаточно полное трёхмерное моделирование салона и расчёт положения в нём робота весьма трудоёмки и времязатратны, что противоречит требованию быстрого развёртывания.

- для нахождения элементов управления можно было бы использовать цветовые и/или текстурные характеристики их поверхностей; решить эту задачу позволяют многочисленные известные алгоритмы сегментации через кластеризацию. Однако с помощью цветовых характеристик трудно учесть беспорядочные перепады яркости, характерные для условий задачи, а текстурные признаки более устойчивы к колебаниям освещённости, но гораздо менее специфичны: текстуры поверхностей у деталей одного салона часто совпадают. Поэтому применить данные алгоритмы как составные части распознавателя можно, но полагаться исключительно на них опасно; желательно организовать проверку данных.

- для удержания захваченных органов управления можно применить в сочетании с визуальной тактильной информацией сенсоров робота при их наличии; при этом для одних органов управления (например, рычага) эти виды информации должны полностью заменять друг друга, для других (например, руля) – дополнять.

Как видно, все рассмотренные подходы обладают значительными недостатками применительно к рассматриваемым условиям. Однако можно решить поставленную задачу, если разместить хорошо различимые для системы визуального распознавания метки на элементах управления. Так

как метки должны размещаться и убираться достаточно быстро, не требовать квалифицированного обслуживания и точной установки, то наиболее простым вариантом таких меток являются бумажные наклейки. Для того чтобы их можно было уверенно различать при колебаниях освещённости, на наклейки нанесена текстура в виде полос или клеток с шагом, допустимые размеры которого определяются угловым разрешением видеокamеры робота.

### **3. Алгоритм распознавания органов управления**

Для распознавания текстурных меток на руле целесообразно использовать распространённый детектор Лоу (Law). Данный алгоритм для каждой точки получает серию величин, измеряющих распределение энергии двумерного сигнала-изображения по пространственным частотам. Вычисление проводится путём обработки предварительно сглаженного изображения серией линейных фильтров с подобранными частотными характеристиками. Полученные величины (в данном случае использовался вариант с 9 мерами [Ertuğrul, 2014]) служат признаками для дальнейшей классификации. В качестве классификатора точек были испробованы различные алгоритмы, но в конечном счёте оказалось достаточно использовать нормальный байесовский классификатор, обученный на случайно сгенерированной выборке точек из размеченной вручную последовательности кадров.

Полученные в результате обработки данные были использованы сначала для поиска меток на изображении. При этом предполагается, что на этом этапе метки достаточно хорошо видны и руки робота не загораживают никакую часть руля. Метки ищутся в виде сплошных областей, которые содержат текстуру метки по результатам работы классификатора; эти области формируются в явном виде после порогового преобразования и применения морфологической операции эрозии.

Положение руля в пространстве можно оценить, если камера, которая проводит съёмку, откалибрована, то есть допускает вычисление в системе координат камеры положения луча, проходящего через оптический центр камеры и заданную точку изображения с неё. В используемом роботе калибровка камеры производится встроенными средствами его операционной системы; при необходимости можно было бы проделать эту операцию и самостоятельно с помощью калибровочных шаблонов с известными параметрами. В текущей версии алгоритма для минимизации объёма вычислений используется только одна из двух головных камер робота.

При размещении меток на руле необходимо измерить попарные расстояния между ними. Тогда метки и центр камеры образуют

треугольную пирамиду, причём в ней известны 3 угла, прилегающие к вершине-оптическому центру, и 3 ребра, лежащие против каждого из углов. Для определения положения камеры необходимо определить длины неизвестных рёбер пирамиды. Решив данную задачу, мы получаем текущее положение рулевого колеса относительно головы робота; при наличии данных о положении его суставов и о геометрии робота можно определить и глобальные координаты руля.

Начальное положение рычага по данным одной камеры и без знания его размеров найти однозначно невозможно, но можно, действуя аналогично прошлому случаю, найти луч, на котором лежит рычаг. В дальнейшем манипулятор робота должен быть выведен на траекторию, соответствующую этому лучу.

Изложенные действия позволяют найти положение рулевого колеса, если ни одна из меток не заслонена рукой, и все они хорошо распознаны. Однако этого совершенно недостаточно для обеспечения устойчивой работы алгоритма в практически важных случаях. На рис. 2 справа приведён пример изображения, когда руль при быстром повороте заслоняется рукой экспериментатора, причём метки при данной частоте работы камеры не могут быть выявлены текстурным анализом. Для того, чтобы суметь при этом распознать рулевое колесо, необходимо дополнить алгоритм следящим компонентом.



Рис. 2. Примеры изображения руля с метками в разных ситуациях

Для слежения за рулём, независимого от текстурных меток, необходимо сравнивать имеющихся в текущий момент данные с полученными ранее в лучших условиях сведениями о внешнем виде руля. В качестве признака, определяющего руль, было решено использовать интегральное распределение яркости в кольцевой области, занимаемой на изображении рулевым колесом. Общая схема алгоритма поиска руля приведена на рис. 3.

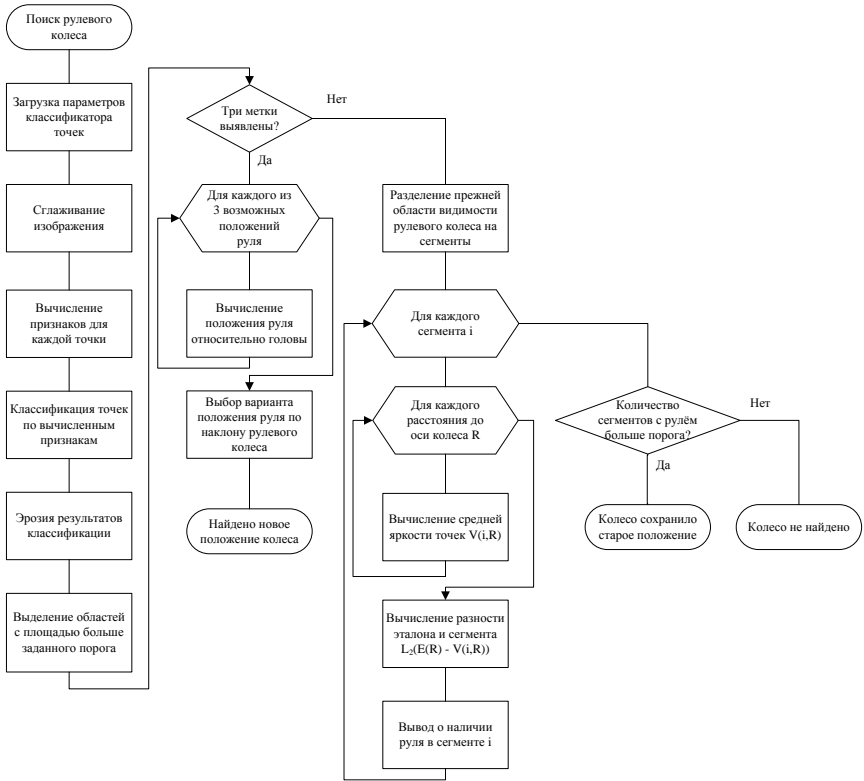


Рис. 3. Схема алгоритма поиска руля на изображении

Эталон яркости формируется при первом распознавании руля. В дальнейшем при потере видимости хотя бы одной из меток яркость области, в которой руль наблюдался в последний раз, сравнивается с эталоном. Для вычисления результатов такого сравнения кольцевая область поиска руля делится на сектора (рис. 4), и по каждому из них отдельно вычисляется разность эталона  $E$  и наблюдаемой яркости  $V$ . Если эта величина превосходит заданный порог  $D$ , считается, что в данной части кольца руль не виден. Далее производится голосование отдельных элементов кольца с задаваемым порогом принятия утверждения о том, что руль наблюдается в проверяемом месте. В случае, если результат голосования отрицателен, считается, что руль потерян из зоны видимости.



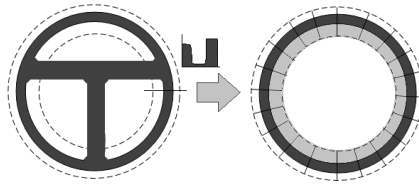


Рис.4. Формирование эталона яркости для слежения за рулём

#### 4. Экспериментальная проверка работы системы

Реализация алгоритма распознавания была выполнена в виде модуля ROS с применением языков программирования Python и C++. Работа с изображениями и видеоданными осуществлялась с применением библиотеки OpenCV. Примеры результатов работы распознавателя руля в разных ситуациях приведены на рис.5.

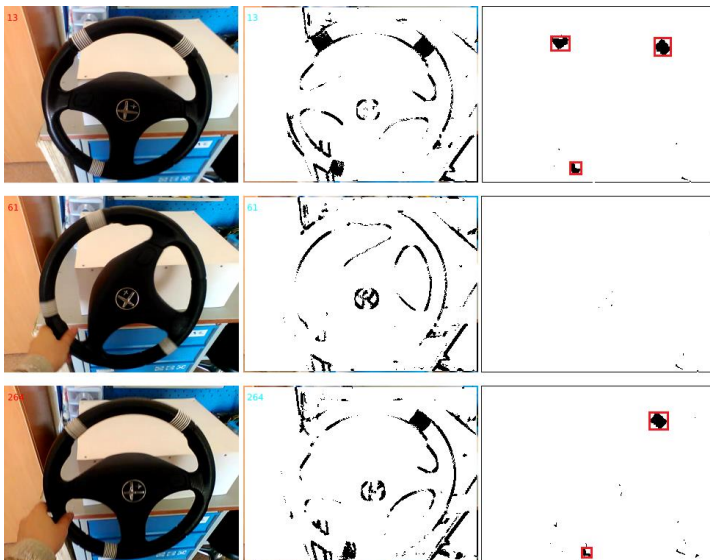


Рис. 5. Примеры результатов работы алгоритма распознавания текстур. Для трёх разных кадров показаны: необработанные данные с камеры (слева); результат классификации точек, где чёрным цветом обозначены точки, распознанные как точки меток (в центре); результат поиска меток после преобразования эрозии (справа)

Видно, что на верхнем кадре успешно распознаны все три метки. На среднем (при быстром вращении руля) и нижнем (при медленном

вращении) изображениях распознано меньше трёх меток, и следящая система в этих случаях сохраняет прежнее положение рулевого колеса. Для обучения классификатора текстур использована размеченная тестовая видеозапись (430 кадров, 20 000 примеров, текстуры в виде чёрных полос либо клеток на белом фоне). Доля успешно распознанных байесовским классификатором точек меток составила 84%. Как видно на рис. 5, имели место многочисленные ложные срабатывания классификатора вне меток на границах однородных областей изображений, что потребовало применения операции эрозии к результатам классификации.

Для проверки точности определения положения рулевого колеса была проведена запись его плавного вращения на жёстко закреплённую в известном, относительно колеса, положении видекамеру (расстояние до центра передней поверхности колеса 30 см, камера направлена на центр колеса, угол наклона камеры и оси колеса одинаков). Далее для каждого кадра была определена линейная ошибка вычисленного положения центра колеса в направлении съёмки и в перпендикулярном ему направлении. Итоговые результаты для 100 проанализированных кадров приведены на рис.6. Из результатов эксперимента можно сделать вывод, что ошибка определения положения рулевого колеса в благоприятных условиях не превышает 2.5 см; это позволяет использовать систему для антропоморфных роботов в натуральную человеческую величину.

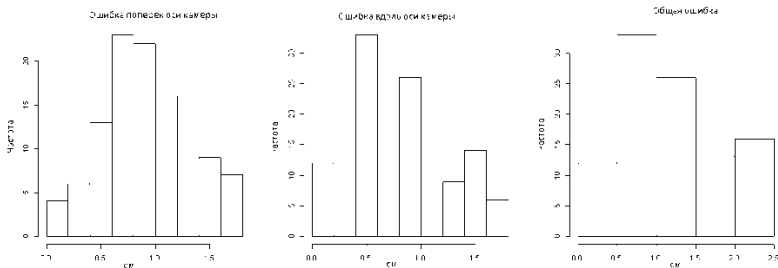


Рис. 6. Ошибки определения положения рулевого колеса

Для проверки следящего алгоритма проведено тестирование на 3 сериях по 20 последовательных изображений руля, получаемых сдвигом полных изображений за границу кадра с заполнением пустого пространства равномерным серым цветом. При этом во всех изображениях после первого одна из меток была закрасена цветом руля. В каждом случае фиксировался кадр, начиная с которого руль не будет распознан. Для всех серий видимая доля руля, при которой он считался распознанным, совпала с порогом принятия решения нечёткой системой.

Свойства полученного алгоритма позволяют использовать его на учебной модели антропоморфного робота лаборатории, предназначенной для проектной деятельности школьников в области мобильной робототехники [Сорокоумов, 2017].

## Заключение

Результатом разработки стала система компьютерного зрения, позволяющая антропоморфному роботу определять на видеопотоке органы управления автомобилем. Данная разработка будет использована для проведения экспериментов по моделированию системы управления движением конечностей человека, но может быть применена и для более общих задач, требующих распознавать объекты в условиях переменной освещённости.

## Список литературы

- [Ackerman,2015] Ackerman E., Guizzo E. DARPA Robotics Challenge: Amazing Moments, Lessons Learned, and What's Next. IEEE Spectrum, June 11, 2015. // Электронный ресурс: <http://spectrum.ieee.org/automaton/robotics/humanoids/darpa-robotics-challenge-amazing-moments-lessons-learned-whats-next>
- [Atkeson, 2015] Atkeson C. G., Babu B.P.W., Banerjee N. et al. No falls, no resets: Reliable humanoid behavior in the DARPA robotics challenge. IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), 2015. DOI: 10.1109/HUMANOIDS.2015.7363436
- [Herault, 2010] Herault J. Vision: Images, Signals and Neural Networks-Models of Neural Processing in Visual Perception. WorldScientific, 2010.
- [Ertuğrul, 2014] Ömer Faruk Ertuğrul. Adaptive texture energy measure method. International Journal of Intelligent Information System, 2014, no 3(2), pp. 13-18.
- [Pfeifer, 2012] Pfeifer R., Lungarella M., Iida F. The Challenges Ahead for Bio-inspired 'Soft' Robotics. Communications of the ACM, November 2012, vol. 55, no. 11. DOI:10.1145/2366316.2366335
- [Rosenzweig, 2015] Rosenzweig J., Bartl M. A Review and Analysis of Literature on Autonomous Driving. The Making of Innovation, October 2015 // Электронный ресурс: [http://www.michaelbartl.com/co-creation/wp-content/uploads/Lit-Review-AD\\_MoI.pdf](http://www.michaelbartl.com/co-creation/wp-content/uploads/Lit-Review-AD_MoI.pdf)
- [Reem-C] REEM-C: робототехнические исследования // Электронный ресурс: <http://pal-robotics.com/ru/products/reem-c/>
- [Shapiro, 2001] Shapiro L., Stockman G. Computer Vision. Prentice–Hall, 2001.
- [Сорокоумов, 2017] Сорокоумов П.С. Учебный проект робота, управляющего автомобилем. Открытое образование. 2017;(2):4-13. DOI:10.21686/1818-4243-2017-2-4-13

УДК 004.896:621.865

## ГРАФЫ КАК ИНСТРУМЕНТ ДЛЯ РЕШЕНИЯ ЗАДАЧИ РАСПОЗНАВАНИЯ СЦЕН

А.Д. Московский (*moscowskyad@gmail.com*)  
НИЦ «Курчатовский институт»

**Аннотация.** В работе рассматривается метод распознавания сцен как часть общей системы технического зрения. В качестве инструмента для решения данной задачи предлагается использовать графовый механизм, т.к. данное описание естественно для представления сцен. Также в работе применен метод недоопределенных моделей для разрешения ситуаций с неполными входными данными.<sup>1</sup>

**Ключевые слова:** распознавание, техническое зрение, реконструкция сцен, недоопределенные модели, доопределение атрибутов.

### Введение

В работе рассматривается вопрос распознавания и анализа сцен. Распознавание сцен является классической задачей искусственного интеллекта, и является крайне актуальной для робототехники, прежде всего - мобильной. Под анализом трехмерной сцены (по соответствующему ей двухмерному изображению) изначально понималась задача выделения на изображении областей, принадлежащим разным объектам и фону, а также разбиение сложных объектов на составные части. Одними из первых работ на эту тему были [Roberts, 1965], [Guzman, 1967] и [Huffman, 1971]. В основном это были первые шаги по объединению простых объектов в более сложные структуры. Важным этапом стала работа [Marvin Minsky, 1975], где помимо самих объектов рассматривались отношения между ними. Такое представление было введено для решения задачи распознавания сцен субъектом, находящимся в движении. С тех пор в литературе обычно под понятием сцена подразумевается набор объектов и набор отношений между объектами [Карпов, 2011]. Несмотря на развитие методов получения трехмерных данных по видеокамере с использованием стереозрения [Scharstein, 2001] или structure-from-motion [Jebara, 1999], а так же появления таких датчиков как Kinect, распознавание сцен является по-

---

<sup>1</sup> Работа выполнена при частичной поддержке гранта РФФИ 15-07-07483

прежнему актуальной задачей. Из интересных работ можно привести метод [Brown, 2011], использующий алгоритм SIFT для задачи классификации сцен. Также распознавание типов сцен по характерным признакам рассматривается в работе [Parizi, 2012], а алгоритм, позволяющий работать с большими объемами распознаваемых паттернов сцен представлен в [Zhou, 2014]. Одной из проблем распознавания сцен, особенно методами получения трехмерных данных, является слабая возможность работы с динамическими сценами. Существует несколько исследований, описывающих работу со сценами, изменяющимися во времени, см. например [Yu, 2011]. Однако в большинстве работ этот вопрос остается не рассмотренным, в то время как данный критерий является крайне важным в задачах мобильной робототехники.

В настоящей работе предлагается метод распознавания сцен, с применением элементов теории графов и механизма недоопределённых моделей. При этом будет предполагаться, что задача поиска отдельных объектов, из которых формируются сцены, является решенной, так, как это описано в [Московский, 2016].

## 1. Постановка задачи

Пусть имеются данные от системы распознавания, представляющие собой информацию о найденных объектах. Также есть начальные условия, сообщающие системе описания всех сцен, которые требуется распознать. Как говорилось выше, описание сцены представляет собой набор объектов и набор отношений между ними. Отношения подразумеваются в первую очередь пространственные («быть над», «быть под», «быть на заданном расстоянии») и сравнительные («быть больше», «быть меньше»). Задать сцену можно посредством матрицы, где номера строк и столбцов соответствуют объектам в базе, а элементы матрицы соответствуют отношениям (при условии, что объекты идентифицированы и упорядочены).

	O <sub>1</sub>	O <sub>2</sub>	O <sub>3</sub>	O <sub>4</sub>
O <sub>1</sub>	-	R <sub>2</sub>	R <sub>6</sub>	R <sub>1</sub>
O <sub>2</sub>	R <sub>2</sub>	-	R <sub>3</sub>	R <sub>5</sub>
O <sub>3</sub>	R <sub>6</sub>	R <sub>3</sub>	-	R <sub>4</sub>
O <sub>4</sub>	R <sub>1</sub>	R <sub>5</sub>	R <sub>4</sub>	-

Рис. 1. Матричное описание сцены

Задачей является сопоставить текущий наблюдаемый набор с одной из заранее заданных сцен. С одной стороны существует ряд способов решения

данной задачи, с другой с учетом особенностей мобильной робототехники, задача приобретает ряд сложностей, связанных с проблемами визуальных сенсоров и помех создаваемыми окружающей средой. К этим помехам можно отнести заслонение объектов, неполную освещённость, неудачный ракурс и различные шумы. В связи с этим информация об объектах собирается частичная, не полная, но разрешаться эта проблема может различными путями, например, с помощью недоопределённых моделей, введённых Нариньяни [A.S.Narin'yanı, 1997]. Предложенный в [Московский, 2016] метод позволяет строить предположения о нахождении объектов на изображении, когда входная информация неполная или искаженная. Объекты, поступающие на вход алгоритма, помимо своих естественных характеристик (положение на изображении, тип объекта) содержат еще и коэффициент уверенности, сообщаящий насколько точно объект был обнаружен в терминах системы. Поскольку информация об объектах может подаваться также в неполном виде, то метод распознавания сцен требуется также снабдить механизмом недоопределенных моделей, что бы повысить его эффективность.

## 2. Описание подхода

### 2.1 Описание сцены

Формализуем описание сцены. Сцена задается набором объектов

$$O(t) = \{o_1, o_2, \dots\} \quad (1)$$

и набором отношений

$$R(t) = \{r_1^2, r_i^j, \dots\} \quad (2)$$

Сцены, которые представлены в базе, и те которые требуется распознать, будем называть эталонными сценами. Помимо матричной формы задания сцен, удобно представлять сцену в виде графа, где вершинами являются объекты, а ребрами – отношения.

$$G = (O, R) \quad (3)$$

По каждой эталонной сцене строится свой граф  $G_{ref}$ . Этот граф является неориентированным и обязан быть связным.

### 2.2 Распознавание сцен

На каждом шаге алгоритма на вход поступает ряд распознанных объектов, а также отношений, регистрируемых между ними. Назовем это текущей сценой. Текущая наблюдаемая сцена представляется также в виде графа  $G_{cur}$  путем добавления всех объектов и отношений на него. Таким образом, мы получаем граф, содержащий полную информацию о

наблюдаемой сцене. Далее требуется распознать, присутствуют ли среди данного графа какие-либо подграфы, соответствующие эталонным сценам. Это все сводится к естественной задаче поиска подграфа. Однако в силу определенной выше специфики задачи, информация может быть неполная, т.е. могут быть неправильно определены или вообще отсутствовать некоторые объекты и отношения на данной сцене. Также из-за того, что объекты на изображении естественным образом разрознены, то граф текущей сцены не обязан быть связным. Поэтому прежде чем решать задачу поиска подграфа, требуется произвести предобработку графа.

Первый шаг предварительной обработки графа заключается в разбиении графа на связные подграфы.

$$G_{ref} \rightarrow \{G_1, G_2, G_3, \dots, G_k\} \quad (4)$$

Данная процедура нужна, чтобы определить базовые структуры сцены, от которых можно оттолкнуться при дальнейшем анализе. Далее каждый из связных подграфов требуется сравнить с графами эталонных сцен. Связный подграф может иметь совершенно произвольное количество элементов, зависящее от условий наблюдения и наблюдаемой сцены, в то время как граф эталонной сцены всегда фиксирован. Перед тем как определять похожесть, надо учесть все возможные варианты, которые может представлять связный подграф текущей сцены. Предполагается, что на текущей сцене обычно находится больше объектов одного типа, чем должно быть, поэтому будем искать эталонный граф как подграф текущего. Однако на текущем графе могут и отсутствовать часть объектов, составляющих эталонную сцену. В связи с этим создается временный граф (для каждого связного подграфа текущей сцены), который получается путем удаления с эталонного графа вершин объектов, не представленных на текущем связном подграфе. Соответственно удаляются и все связные с этими вершинами ребра.

$$G_{temp} = G_i \setminus \{o_p \dots o_k\} \quad (5)$$

Эта операция гарантирует, что текущий граф будет содержать меньше или равное связному подграфу текущей сцены количество вершин. После того как временный граф подготовлен, можно выполнить его поиск на соответствующем ему связном подграфе. Это задача поиска изоморфизма подграфов. Среди многих решений данной задачи, был выбран алгоритм VF2 и его реализация на языке C++, представленная в библиотеке `igraph` [Документация `igraph`, 2017]. Алгоритм VF2 возвращает карту соответствий вершин временного графа и связного подграфа текущей сцены. По этой карте можно отобрать объекты, формирующие эталонную сцену на текущей сцене.

$$VF2(G_i, G_{temp}) = G_{wanted} \quad (6)$$

Описанный процесс позволяет подготовить подграф для задачи поиска изоморфизма подграфов в случае потерянных объектов. Если отношения потеряны, то данная схема не даст положительных результатов. Чтобы решить эту проблему, предлагается преобразовать эталонный граф и граф текущей сцены в их реберное представление. Это представление меняет ребра и грани местами. Имея такое представление можно применить описанную выше процедуру.

Если изоморфный подграф был найден, и на временном графе не пришлось удалять объекты, то задача распознавания сцены была бы решена. Однако в случае с отсутствием некоторых объектов требуется проводить дальнейшую обработку результатов алгоритма поиска изоморфных подграфов.

### 2.3 Доопределение элементов сцены

Наиболее интересен случай, когда не было распознано несколько отношений, но сами объекты присутствуют на текущей сцене (предположим, из общей картины выпало столько отношений, что один объект не попал на связный подграф). В таком случае механизм, описанный выше, возвращает те объекты на изображении, которые соответствуют эталонной сцене, но без потерянного объекта. Далее выполняется поиск данного объекта, и тут возникает та самая неопределенность. Изолированных от сцены объектов может быть несколько, и каждый потенциально может являться частью сцены, но поскольку отношения для него распознано не было, то нельзя определить какой именно (или же ни одного из представленных). Для статических задач распознавания на отдельном изображении эта задача не очень простая и требует дополнительной входной информации, например, контекста. Однако в данной работе предполагается использование метода в робототехнических системах, что позволяет говорить о распознавании на последовательности изображений. Также как в случае распознавания объектов, предлагается прибегнуть к механизму недоопределённых моделей (n-моделей) [Нариньяни, 1998]. Напомним, что в случае с объектами, система отслеживает их перемещение от кадра к кадру и, таким образом, позволяет определить соответствия между объектами на разных стадиях наблюдения.

Обобщенная вычислительная модель данной методологии задается четырьмя множествами:

$$M=(V,W,C,R) \quad (7)$$

где  $V$  – множество всех объектов,  $W$  - множество функций присваивания,  $C$  - множество функций проверки корректности,  $R$  - множество ограничений. Объектами множества  $V$  в рассматриваемом случае являются объекты,



получаемые от системы распознавания. Функциями присваивания  $W$  выступает в данном случае механизм, относящий объекты к эталонным сценам. Функции  $S$  выполняют проверку корректности выбранных изолированных объектов в соответствии с ограничениями  $R$ . Смысл представленной вычислительной модели - в итерационном применении данных функций к объектам. Функции и ограничения должны задаваться таким образом, что бы итерационное применение как минимум не расширяло область неопределенности, однако сходимость метода не гарантируется.

Ограничения заданы таким образом, что в сцену попадает только тот изолированный объект, который уже был к ней причастен на предыдущих шагах распознавания. Этот подход позволяет поддерживать представление о сцене в случае ухудшения условий наблюдения. Этот же подход работает аналогичным образом, если изолированных объектов несколько в сцене.

Также интересен случай, когда нераспознанные отношения разбивают сцену на две части, и каждая часть при этом выступает как распознанная часть эталонной сцены. Данный случай разрешается также как с изолированными объектами, только добавляется проверка на не повторение одних и тех же объектов в общей сцене.

После того, как все распознающие процедуры были произведены, для каждой сцены требуется рассчитать коэффициент доверия, подобно тому, как это делается для объектов.

$$CC_S = \frac{1}{N_{ratio}} \left( \sum_{ratio} \frac{1}{2} (CC_{Oi} + CC_{Oj}) + \sum_{isolated} \frac{CC_{Ok}}{N_{con.ratio} + 1} \right) \quad (8)$$

Первое слагаемое коэффициента уверенности представляет собой сумму по всем распознанным отношениям полусумм коэффициентов уверенности связанных отношением объектов к общему числу отношений. Второе слагаемое - это сумма по всем изолированным объектам их коэффициентов уверенности к количеству исходящих отношений из данного объекта плюс один. Вся сумма делится на общее число отношений в сцене. Если все отношения и объекты были распознаны, то коэффициент уверенности становится равным единице.

Общая схема, описанного в работе метода представлена на рисунке 2.



Рис. 2. Общая схема предложенного метода распознавания сцен

### 3. Эксперименты

Исследование работоспособности данного метода производилось следующим образом. Генерировался граф с заданным количеством вершин и случайными связями, это была эталонная сцена. Далее этот граф подвергался различному искажению: случайно удалялись или добавлялись вершины и грани. Полученный граф играл роль наблюдаемой сцены. Далее для полученных графов применялся описанный в работе метод. На рис. 3 представлена зависимость коэффициента доверия (8) от искажений, внесенных в граф.

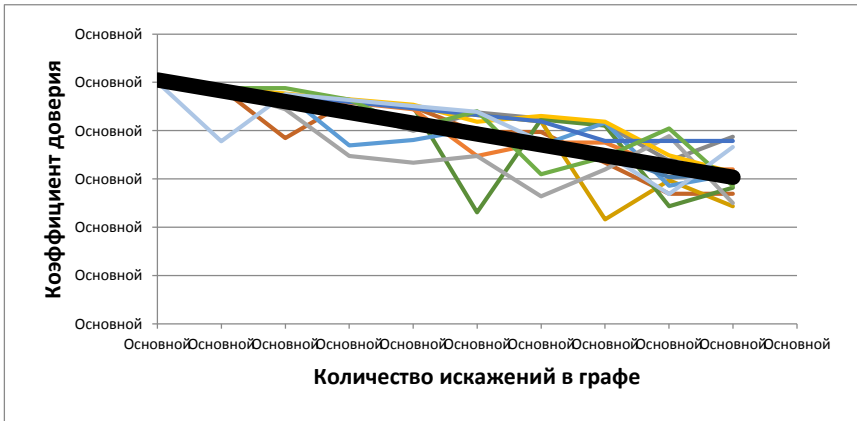


Рис. 3. Зависимость коэффициента уверенности от количества искажений для набора графов с 20-25 вершинами, черной жирной линией указана линейная аппроксимация

Из графика видно, что метод достаточно стабильно справляется с распознаванием, даже когда граф отличается почти на половину элементов (коэффициент доверия в районе 0.5).

## Заключение

Был разработан метод распознавания сцен для мобильных робототехнических систем, использующий графовый механизм и Н-модели. Проведенные эксперименты показали не только работоспособность подхода, но также достаточную скорость работы алгоритма для использования в бортовых системах мобильных роботов. При этом приемлемая скорость обработки (2-5 Гц) достигается на выборках, содержащих не более ста объектов на текущей сцене.

## Список литературы

- [Карпов, 2011] Карпов В.Э. Об одном механизме реконструкции схем // VI-ая Международная научно-практическая конференция «Интегрированные модели и мягкие вычисления в искусственном интеллекте», Сб. науч. трудов. В 2-т., Т.1, М.: Физматлит. , 2011.
- [Московский, 2016] Московский А.Д. Об одном методе распознавания объектов с не полностью определенными признаками // Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2016), Казань. 2016. С. 137–146.
- [Нариньяни, 1998] Нариньяни, А.С., Телерман, В.В., Ушаков, Д.М., Швецов, И.Е. Программирование в ограничениях и недоопределенные модели // Информационные технологии. 1998. Т. 7.
- [A.S.Narin'yani, 1997] A.S.Narin'yani, S.B.Borde, D.A.Ivanov. Sub-Definite Mathematics and Novel Scheduling Technology Programs // Artif. Intell. Eng. 1997. Т. 11.
- [Brown, 2011] Brown M., Susstrunk S. Multi-spectral SIFT for scene category recognition // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. , 2011. С. 177–184.
- [Guzman, 1967] Guzman A. Some Aspects of Pattern Recognition by Computer // MAC-TR-37. 1967.
- [Huffman, 1971] Huffman D. Impossible Objects as Nonsense Sentences // Mach. Intell. 6. 1971.
- [Jebara, 1999] Jebara T., Azarbayejani A., Pentland A. 3D structure from 2D motion // IEEE Signal Process. Mag. 1999. Т. 16. № 3. С. 66–84.
- [Marvin Minsky, 1975] Marvin Minsky. A Framework for Representing Knowledge // Psychol. Comput. Vis. 1975.
- [Parizi, 2012] Parizi S.N., Oberlin J.G., Felzenszwalb P.F. Reconfigurable models for scene recognition // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. , 2012. С. 2775–2782.

- [**Roberts, 1965**] Roberts L. Machine Perception of Three Dimensional Solids / под ред. Т. J. Cambridge: , 1965. Вып. The MIT Pr.
- [**Scharstein, 2001**] Scharstein D., Szeliski R., Zabih R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms // Proceedings - IEEE Workshop on Stereo and Multi-Baseline Vision, SMBV 2001. , 2001. С. 131–140.
- [**Yu, 2011**] Yu, X., Fermuller, C., Teo, C. L., Yang, Y., & Aloimonos, Y. (2011). Active scene recognition with vision and language. In 2011 International Conference on Computer Vision, ICCV 2011. Pp. 810-817.
- [**Zhou, 2014**] Zhou B., Khosla A., Lapedriza A. Learning Deep Features for Scene Recognition using Places Database // Adv. Neural Inf. Process. Syst. 27. 2014. С. 487–495.
- [**Документация igraph, 2017**] Документация igraph [Электронный ресурс]. URL: <http://igraph.org/c/#docs> (дата обращения: 14.05.2017).

УДК 681.786

## СРАВНЕНИЕ СИСТЕМ КООРДИНАТНЫХ МЕТОК ДЛЯ КАЛИБРОВКИ КАМЕР МОБИЛЬНОГО РОБОТА В УСЛОВИЯХ ПЕРЕКРЫТИЙ

К.С. Шабалина (*shabalina@it.kfu.ru*)

А.Г. Сагитов (*sagitov@it.kfu.ru*)

Е.А. Магид (*magid@it.kfu.ru*)

Высшая Школа Информационных Технологий и  
Информационных Систем, Казанский (Приволжский)  
Федеральный Университет, Казань

**Аннотация.** Система координатных меток – система уникальных 2D-меток, которые помещаются в окружающую среду и могут быть автоматически обнаружены камерой с помощью соответствующего алгоритма. Системы меток необходимы для осуществления калибровки камер и манипуляторов робототехнических систем с возможностью калибровки в автономном режиме. При этом, система должна быть устойчива к таким факторам как угол обзора, полное или частичное перекрытие метки, изменение расстояния до метки и др. Эта статья сравнивает три существующие системы меток: ARTag, AprilTag и CALTag. Основная цель работы - исследование надежности распознавания метки в условиях перекрытия разных типов и уровня, которые часто встречается на практике, затрудняя калибровку камер и манипуляторов мобильных роботов.<sup>1</sup>

**Ключевые слова:** координатные метки, сравнительный анализ, окклюзия, ARTag, AprilTag, CALTag, мобильный робот.

### Введение

Координатные метки (англ. fiducial markers) являются основными элементами в отслеживающих системах, которые используются в робототехнике и приложениях дополненной реальности для стабилизации поля зрения. Для упрощения процесса отслеживания, выравнивания и идентификации координатные метки крепятся к объектам. В робототехнике

---

<sup>1</sup> Работа осуществлена при финансовой поддержке РФФИ и Правительства Республики Татарстан в рамках научного проекта № 17-48- 160879. Часть работ выполнена в соответствии с Государственной программой конкурентоспособности Российской Федерации при Казанском федеральном университете.

метки позволяют калибровать камеру и механические части в робототехнических системах (РТС), которые востребованы в промышленных целях [Klimchik et al., 2016], взаимодействию РТС и человека, разработке антропоморфных РТС [Khusainov et al., 2015], методах одновременной локализации и построения карты (англ. SLAM), спасательной робототехнике [Magid et al., 2010] и других областях. При этом для беспилотных РТС с искусственным интеллектом операции калибровки камер и манипуляторов должны осуществляться в автономном режиме по мере востребованности системой управления.

Координатные метки существуют во многих вариациях: например, простые формы из нескольких точек, штрих-коды в форме эллипса, прямоугольные изображения. Использование четырех угловых точек прямоугольных маркеров позволяет однозначно определить координатное положение и ориентацию маркера относительно уже откалиброванной камеры. Метки должны быть устойчивы к таким факторам, как угол обзора камеры, окклюзии, изменение расстояния от камеры и др.

Для наших исследований координатные метки (или так называемые «теги») представляют интерес для калибровки трех передних (стереопара и моно-камера) и одной задней камер (моно-камера), и последующей калибровки двухзвенного манипулятора мобильного РТС «Сервосила Инженер» [Sokolov et al., 2017], а также для калибровки одной передней и одной задней камер (моно-камеры), и последующей калибровки двух двухзвенных манипуляторов типа «рука» антропоморфного РТС AR-601M [Khusainov et al., 2015]. Прежде чем приступить к калибровке на основе одной из систем меток, необходимо выбрать оптимальный вариант системы под наши требования. В этой работе проводится детальное сравнение тегов ARTag, AprilTag и CALTag на устойчивость к окклюзии. Для сравнения маркеров была применена окклюзия разных типов и уровня интенсивности. Контрольным показателем в работе является устойчивость и способность обнаружения маркера системой при наличии окклюзии разного характера и интенсивности. В работе предоставлено экспериментальное сравнение трех вышеуказанных маркеров. Обнаружение маркеров было выполнено с помощью недорогой веб-камеры.

## 1. Координатные системы маркеров

Большинство маркеров имеют общую особенность дизайна – внешняя квадратная форма с паттерном внутри, в котором закодирована информация. Квадратная форма популярна в силу того, что это позволяет обнаружить как минимум четыре особые точки (соответствующие точкам углов квадрата), которые позволяют калибровать камеру и рассчитать позицию маркера. Чтобы уменьшить чувствительность к световым

условиям, большинство маркеров сделаны монотонными [Hirzer, 2008]. ARTag, AprilTag, и CALTag системы имеют две стадии обнаружения тегов: поиск и детектирование уникальных особенностей (англ. unique features) и распознавание тега [Fiala, 2005a]. На первом этапе системой обнаруживаются особенности тега - квадратная форма и четыре угловые точки. На втором этапе проверяется внутреннее изображение для определения, является ли обнаруженный объект тегом или нет. Различный дизайн маркеров, алгоритмы обнаружения и распознавания влияют на сильные и слабые стороны тегов в различных ситуациях. Для оценки тегов существует несколько критериев: устойчивость к окклюзии (полное или частичное перекрытие тега другим объектом), устойчивость к изменению освещения, вероятность ошибочного принятия одного маркера за другой, минимальные размер маркера (или максимальное допустимое расстояние до маркера), и ложный/негативный эффект и другие [Fiala, 2004]. На рисунке 1 изображены примеры маркеров ARTag, AprilTag и CALTag.



Рис. 1. Маркеры ARTag, AprilTag, CALTag (слева направо)

## 1.1 ARTag

ARTag является системой маркеров, изначально созданной для улучшения системы ARToolKit [Fiala, 2004], и использует более сложный процесс распознавания и обнаружения маркера для повышения надежности системы по всем критериям. Изначально предполагалось, что ARTag заменит ARToolKit из-за использования новых алгоритмов распознавания и обнаружения: благодаря новым подходам, эта система успешно решила проблемы ARToolKit – обнаружение маркеров в пространстве при их фактическом отсутствии и ошибочное принятие одного маркера за другой. ARTag использует ту же идею о квадратной форме с внутренним изображением (в данном случае – штрих-кодом), но, в отличие от ARToolKit, используется цифровой подход чтения внутреннего паттерна – бинарного кода. ARTag система состоит из 2002 индивидуальных квадратных маркеров, среди которых 1001 тег имеет черную рамку, и 1001 тег – белую [Fiala, 2005b]. Внутреннее изображение представляет собой штрих-код, состоящий из сетки 6x6 ячеек, заполненных белыми или

черными ячейками, которые представляют собой закодированные 36 бит информации. Первые 10 бит последовательности отвечают за ID маркера, последние 26 бит являются избыточными и используются для обнаружения и коррекции ошибок для обеспечения уникальности четырех возможных ориентаций маркера.

## 1.2 AprilTag

Процесс обнаружения маркера AprilTag состоит из нескольких этапов: поиск линейных отрезков, обнаружение квадратов, вычисление положения и ориентации тега, и затем декодирование штрих-кода [Olson, 2011]. Поиск линейных отрезков (англ. segments) использует аналогичный ARTag подход, после которого последовательность отрезков обрабатывается для формирования квадрата из найденных сегментов. Чтобы закодировать внутренний паттерн тега, AprilTag использует технологию, основанную на лексикографических кодах (англ. Lexicographic code), которые характеризуются двумя параметрами: количество бит кодового слова (в данном случае - внутреннего паттерна) и минимальным расстоянием Хэмминга между двумя любыми кодами [Trachtenbert, 1996]. AprilTag имеет несколько семейств тегов, отличающихся двумя параметрами: количеством битов для кодирования и минимальным расстоянием Хэмминга.

## 1.3 CALTag

После анализа классической калибровки камеры, основанной на использовании шахматной доски, и координатных меток был предложен маркер CALTag, как специально разработанное решение для калибровки камеры [Atcheson et al., 2010]. Эта система также состоит из двух компонентов: непосредственно маркера и алгоритма его обнаружения. Калибровочный маркер используется в калибровочной сетке, которая внешне подобна шахматной доске. Каждый маркер состоит из матрицы  $M \times N$  из черных и белых квадратов, которые заключены в рамку, содержащую строго черные или строго белые пиксели. После первоначального обнаружения потенциальных маркеров они фильтруются и проверяются путем чтения их бинарных кодов. Любые пропущенные калибровочные точки восстанавливаются, поскольку системе CALTag известен шаблон шахматной доски.

## 2. Дизайн экспериментов

Экспериментальная работа была проведена для сравнения устойчивости ARTag, AprilTag и CALTag к окклюзии. Окклюзией в данной работе



определяется полное или частичное перекрытие исследуемого объекта (тега) другими объектами. Все эксперименты проводились при использовании недорогой веб-камеры Genius FaceCam 1000X. Применение камеры низкого класса было выбрано для исследования возможности каждого тега при использовании недорогого оборудования. Чтобы проанализировать влияние окклюзии на распознавание тегов, было выбрано четыре различных тега семейства ARTag, четыре тега семейства AprilTag и два тега семейства CALTag (с размером CALTag 4x4 и 9x6). Проводились два типа экспериментов: систематическая окклюзия и произвольное перекрытие тегов объектом.

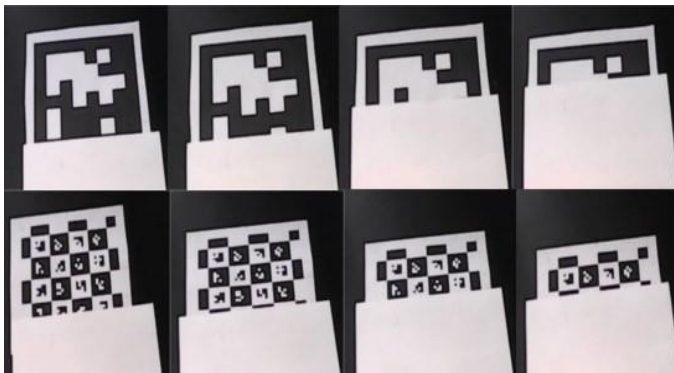


Рис. 2 Систематический подход к окклюзии. AprilTag (верхний ряд) и CALTag (нижний ряд) для  $K=10, 20, 50, 70$  процентов окклюзии (слева направо)

**Тип 1 - Систематическая окклюзия.** Для каждого маркера обеспечивались одинаковые условия освещения помещения и положение камеры относительно метки. Часть каждого тега была перекрыта снизу шаблоном из белой плотной бумаги таким образом, что шаблон перекрывал  $K\%$  от площади маркера, где  $K$  постепенно увеличивалось, принимая значение из 5-значного массива  $[0, 10, 20, 50, 70]$ . На рисунке 2 показаны примеры окклюзии AprilTag и CALTag для  $K=10, 20, 50, 70$  процентов.

**Тип 2 - Произвольное перекрытие тегов объектом.** Каждый тег случайным образом перекрывался одним из трех различных объектов, при этом объект целиком располагался в области тега и поэтому процент перекрытия всегда оставался известным и постоянным. Первым объектом была белая линейка из плотной бумаги, шириной 13 см, длиной 2,5 см, площадью  $32,5 \text{ см}^2$ . Вторым объектом - металлические ножницы площадью  $7,99 \text{ см}^2$ . Третьим объектом - черная пластиковая линейка шириной 15,7 см, длиной 2,6 см, площадью  $40,82 \text{ см}^2$ .

Эксперименты Типа 2 подразделены на две техники перекрытия маркера. Первая заключается в закрытии части внутреннего рисунка тега и пересечения его границ, вторая заключается только в закрытии части внутреннего рисунка тега. Для первой техники объектом был выбрана черная пластиковая линейка, для второй – металлические ножницы и белая линейка. С каждым из трех объектов для каждого тега было проведено 25 экспериментов на каждом теге из сравниваемой системы маркеров (см. рис. 3). Подчеркиваем особый случай для маркера CALTag размера 4x4: если черная линейка и белая линейка расположена строго вдоль стороны маркера, процент занятой области уменьшается по причине того, что ширина обоих объектов превышает размер маркера. В этом случае процент перекрытия варьируется от 25,5% до 33,84% для белой линейки и 25,5% и 33,84% для черной линейки.

### 3. Результаты экспериментов

Для экспериментов применялись официальное программное обеспечение систем AprilTag и CALTag, доступное для публичного использования. Для ARTag использовалась библиотека ArUco, которая позволяет обнаруживать и распознавать различные типы семейств тегов [Garrido-Jurado et al., 2014]. Все теги были распечатаны на белой бумаге следующих размеров: ARTag: 15.2 x 15.2 см, площадь 231.04 см<sup>2</sup>; AprilTag: 13.5 x 13.5 см, площадь 182.25 см<sup>2</sup>; CALTag 4x4: 9.8 x 9.8 см, площадь 96.04 см<sup>2</sup>; CALTag 9x6: 21.7 x 14.7 см, площадь 318.99 см<sup>2</sup>. Идентификаторы ARTag и AprilTag маркеров были выбраны случайным образом. Каждый тег имеет свой собственный уникальный идентификатор, который закодирован во внутреннем рисунке тега. Для экспериментов все AprilTag были выбраны из семейства тегов «36h11», то есть каждый из идентификаторов тегов кодируется в 36-битовом кодовом слове с минимальным расстоянием Хэмминга 11 бит. Для кодирования идентификатора AprilTag использует систему кодирования, основанную на лексикодах [Trachtenbert, 1996].

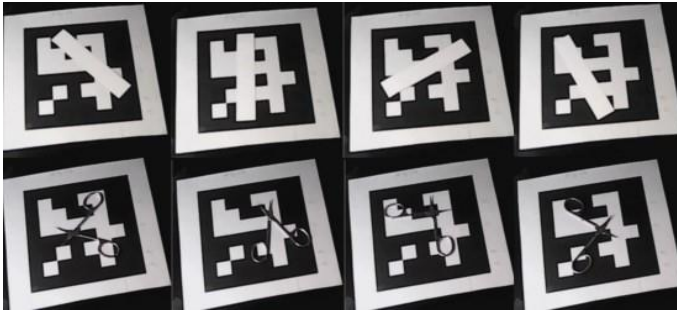


Рис. 3 Произвольное расположение объектов на маркере ARTag: белая линейка (верхний ряд) и ножницы (нижний ряд)

Важно отметить, что два типа окклюзии имеют различную экспериментальную реализацию. Тип 1 отражает типичную ситуацию реального мира, когда объект закрывает и внутренний паттерн, и границы метки. Поскольку границы тегов образуют уникальную особенность тега для его обнаружения, этот тип окклюзии влиял как на обнаружение тега, так и на их распознавание. В Типе 2 объект перекрывал исключительно внутренний рисунок метки, влияя только на этап распознавания тега.

Результаты экспериментов приведены в таблицах 1-4. Таблица 1 демонстрирует результаты экспериментов Типа 1. Значение “1” означает успешное обнаружение тега, “0” означает, что системе не удалось обнаружить тег. ARTag и AprilTag продемонстрировали высокую чувствительность к перекрытию границ тегов: перекрытие границ привело к невозможности обнаружения уникальных признаков (границ), и, соответственно, к отрицательному результату в обнаружении. CALTag оказался единственным устойчивым к такому типу окклюзии маркером, что объясняется его алгоритмом обнаружения [Zhang et al., 2002]. CALTag 9x6 был успешно обнаружен при окклюзии его площади на 50%; CALTag 4x4 - при окклюзии на 70%. Таким образом, система CALTag оказалась более адаптированной к реальным ситуациям с частичной видимостью тега.

Таблица 2 демонстрирует результаты экспериментов Типа 2 с использованием белой линейки. В случае AprilTag только 7 экспериментов из 100 (по 25 на один ID маркера) были успешными: ID4 и ID 6 по одному успешному результату, ID8 - 5, ID9 - 0. ARTag продемонстрировал самые низкие результаты по сравнению с другими системами – всего 2 успешных эксперимента из суммарных 100. Белая линейка затрудняет корректное считывание двоичного кода тегов, так как все теги – монохромные бинарные изображения. В это же время, успешность проведения экспериментов с маркером CALTag 4x4 составила 88%, а CALTag 9x6 -

96%. Эта система тегов более устойчива к перекрытию внутреннего рисунка благодаря дизайну маркера и алгоритму распознавания.

Таблица 3 демонстрирует результаты экспериментов Типа 2 с использованием черной пластиковой линейки в качестве объекта перекрытия. Объект был произвольно помещен во внутреннюю часть тега. Для каждого тега было проведено по двадцать пять экспериментов, чтобы положение объекта на теге было различным в каждом эксперименте. Черный цвет линейки затрудняет чтение двоичного кода тега, потому что все метки являются монохромными и, по условию эксперимента, линейка пересекает контуры меток. Частным случаем в этих экспериментах был тег CALTag 4x4, поскольку из-за его небольшого размера перекрытие области объектом варьировалось от 25,5% до 33,84%. Оба маркера CALTag (4x4 и 9x6) были успешно обнаружены в 46 экспериментах из 50 (92%), в то время как все 50 экспериментов с ARTag и AprilTag оказались неудачными.

Таблица 4 демонстрирует результаты экспериментов Типа 2 с использованием ножниц в качестве объекта перекрытия. Эти эксперименты демонстрируют зависимость алгоритма распознавания тегов при перекрытии только внутреннего рисунка маркера сложным объектом. Маркеры CALTag 9x6 и 4x4 были успешно обнаружены и распознаны во всех 50 экспериментах. Маркер ARTag также продемонстрировал высокую надежность (97%), а маркер AprilTag значительно уступил ему (85%).

*Таблица. 1*

Результаты систематического перекрытия тегов на заданный процент

Тег	0%	10%	20%	50%	70
ARTag (ID 2, 3, 6, 34)	1	0	0	0	0
AprilTag (ID 4, 6, 8, 9)	1	0	0	0	0
CALTag 4x4	1	1	1	1	0
CALTag 9x6	1	1	1	0	0

*Таблица. 2*

Результаты произвольного перекрытия тегов белой линейкой

Тег	Процент перекрытия	Процент успешных экспериментов
ARTag (ID 2)	14.06%	4%
ARTag (ID 3)	14.06%	4%
ARTag (ID 6)	14.06%	0%

ARTag (ID 34)	14.06%	0%
AprilTag (ID 4)	17.83%	4%
AprilTag (ID 6)	17.83%	4%
AprilTag (ID 8)	17.83%	20%
AprilTag (ID 9)	17.83%	0%
CALTag 4x4	22.5% - 32.5%	88%
CALTag 9x6	10.18%	96%

Таблица. 3

Результаты произвольного перекрытия тегов черной линейкой

Тег	Процент перекрытия	Процент успешных экспериментов
ARTag (ID 2, 3, 6, 34)	17.66%	0%
AprilTag (ID 4, 6, 8, 9)	22.39%	0%
CALTag 4x4	26.53% - 40.20%	92%
CALTag 9x6	12.79%	92%

Таблица. 4

Результаты произвольного перекрытия тегов с помощью ножниц

Тег	Процент перекрытия	Процент успешных экспериментов
ARTag (ID 2, 3, 6, 34)	3.45%	97%
AprilTag (ID 4, 6, 8, 9)	4.38%	85%
CALTag 4x4	8.32%	100%
CALTag 9x6	2.5%	100%

## Заключение и планируемые исследования

В работе были подробно описаны три системы маркеров: ARTag, AprilTag и CALTag. Проведены эксперименты с выбранными системами меток для оценивания их чувствительности к различным типам окклюзии. Цель эксперимента состояла в том, чтобы проанализировать слабые и сильные стороны каждого тега в отношении частичной или полной окклюзии маркера. Систематическая окклюзия отражала типичную ситуацию окклюзии реального мира, когда объект закрывает внутреннюю

часть метки и её края. При произвольной окклюзии объектом закрывалось только внутреннее изображение тега, что влияло только на этап распознавания внутреннего паттерна тега. Маркеры ARTag и AprilTag продемонстрировали высокую чувствительность к перекрытию границ тегов, что ограничивает их эффективное использование на практике. Маркеры CALTag показали лучшую устойчивость к окклюзии, как систематической, так и произвольной, включающей перекрытие границ. В ходе экспериментов произвольного перекрытия тегов объектом, ARTag продемонстрировал высокую устойчивость к перекрытию его внутреннего рисунка сложным объектом (ножницами), в то время как у AprilTag выявился более высокий уровень чувствительности к перекрытию при тех же условиях.

В будущем планируется проведение экспериментов на устойчивость тегов к окклюзии с использованием камер различного качества, а также сравнение систем координатных меток по другим критериям оценки. Особое внимание будет уделено CALTag системе, продемонстрировавшей наивысший результат в ходе всех экспериментов. Как долгосрочная цель наших исследований, требуется выбрать наилучшую систему координатных меток, которая будет далее применяться для автоматической калибровки камер и манипуляторов мобильной гусеничной РТС «Сервосила Инженер» и антропоморфной РТС AR-601M.

## Список литературы

- [Atcheson et al., 2010] Atcheson B., Heide F., Heidrich W. CALTag: High Precision Fiducial Markers for Camera Calibration // Vision, Modeling, and Visualization Workshop. — 2010. — vol.10. — С. 41-48.
- [Fiala, 2004] Fiala M. ARTag Revision 1. A Fiducial Marker System Using Digital Techniques // National Research Council Publication. — 2004. — vol. 47419. — С. 1-47.
- [Fiala, 2005a] Fiala M. Comparing ARTag and ARToolKit Plus Fiducial Marker Systems // IEEE Int. Workshop on Haptic Audio Visual Environments and their Applications. — 2005. — С. 148-153.
- [Fiala, 2005b] Fiala M. ARTag, a fiducial marker system using digital techniques // IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. — 2005. — vol. 2. — С. 590- 596.
- [Garrido-Jurado et al., 2014] Garrido-Jurado S., Muñoz-Salinas R., Madrid-Cuevas F.J., Marín-Jiménez M.J. Automatic generation and detection of highly reliable fiducial markers under occlusion // Pattern Recognition. — 2014. — vol. 47(6). — С. 2280-2292.
- [Hirzer, 2008] Hirzer M. Marker Detection for Augmented Reality Applications // Seminar/Project Image Analysis Graz. — 2008. — С. 1-25.
- [Khusainov et al., 2015] Khusainov R., Shimchik I., Afanasyev I., Magid E. Toward a human-like locomotion: Modelling dynamically stable locomotion of an

- anthropomorphic robot in Simulink environment // The 12th Int. Conf. on Informatics in Control, Automation and Robotics. — 2015. — Vol. 2. — C.141-148.
- [Klimchik et al., 2016]** Klimchik A., Magid E., Pashkevich A. Design of experiments for elastostatic calibration of heavy industrial robots with kinematic parallelogram and gravity compensator Pashkevich // 8th IFAC Conf. on Manufacturing Modelling, Management and Control. — 2016 — Elsevier Ltd., vol. 49(12) — C.967–972.
- [Magid et al., 2010]** Magid E., Tsubouchi T. Static Balance for Rescue Robot Navigation: Discretizing Rotational Motion within Random Step Environment // Lecture Notes in Artificial Intelligence. — 2010 — Vol. 6472 — C.423-435.
- [Olson, 2011]** Olson E. AprilTag: A robust and flexible visual fiducial system // IEEE Int. Conf. on Robotics and Automation. — 2011. — C. 3400-3407.
- [Sokolov et al., 2017]** Sokolov M., Lavrenov R., Sagitov A., Sabirova L., Afanasyev I., Magid E. Modelling a crawler-type UGV for urban search and rescue in Gazebo environment // International Conference on Artificial Life and Robotics, — 2017. — C.360-363.
- [Trachtenbert, 1996]** Trachtenbert A. Computational methods in coding theory. MS thesis, University of Illinois at Urbana-Champaign. – 1996.
- [Zhang et al., 2002]** Zhang X., Fronz S., Navab N. Visual Marker Detection and Decoding in AR Systems: A Comparative Study Int. Symposium on Mixed and Augmented Reality. — 2002. — C. 97-106.

УДК 004.896

## ПРОГРАММНЫЙ ИНСТРУМЕНТ ДЛЯ СОЗДАНИЯ 3D-КАРТ В GAZEBO НА ОСНОВЕ ПРОИЗВОЛЬНЫХ ИЗОБРАЖЕНИЙ И ДАННЫХ ЛАЗЕРНОГО СКАНИРОВАНИЯ

А.А. Закиев (*zaufar@gmail.com*)

Р.О. Лавренов (*lavrenov@it.kfu.ru*)

Е.А. Магид (*magid@it.kfu.ru*)

Высшая Школа Информационных Технологий и Систем,  
Казанский Федеральный Университет, Казань

**Аннотация.** Разработка алгоритмов для мобильных роботов, осуществляющих навигацию, построение карт, в том числе, одновременно с локализацией, требует должного моделирования окружающей среды. Наша статья посвящена программному инструменту, который позволяет автоматически создавать реалистичные 3D-ландшафты, основанные на реальных сенсорных данных. Утилита предусматривает фильтрацию карты занятости на входе и импорт её в систему Gazebo в качестве карты высот с возможностью настройки создаваемой в симуляции среды.<sup>1</sup>

**Ключевые слова:** Gazebo; ROS; октокарта; решетка занятости; карта высот; фильтрация карты.

### Введение

Разработка и развитие мобильных робототехнических систем идут очень интенсивно в сегодняшнее время. Проблема навигации – одна из основных трудностей, встающих перед мобильными роботами. Важными частями задачи навигации автономного устройства являются локализация, построение карты и планирование маршрута [Fuentes-Pacheco et al., 2015]. Локализация отвечает за точное определение нынешней позиции робота в окружающем пространстве. Построение карты происходит посредством

---

<sup>1</sup> Работа поддержана Российским фондом фундаментальных исследований (РФФИ) и Министерством науки и технологий и Государством Израиль (совместный проект ID 15-57-06010). Часть работ выполнена в соответствии с Государственной программой конкурентоспособности Российской Федерации при Казанском федеральном университете.



сбора сенсорных данных робота и сохранением их в удобной для последующей обработки форме. Используя данные, полученные в ходе построения карты, становится возможным планирование маршрута, позволяющего из стартовой позиции достигнуть цели, не допуская столкновения с препятствиями. Метод одновременной локализации и построения карты (SLAM) объединяет два процесса для более эффективного обеспечения автономной навигации. Алгоритмы SLAM способны справляться с различными ограничениями аппаратного или внешнего характера, например, SLAM методы для монокулярных систем [A. Buval et al., 2016], SLAM методы, использующие одновременно лидары и визуальные сенсоры [R. W. Wolcott et al., 2014].

Навигационные алгоритмы должны быть тщательно проверены перед их интеграцией в ПО реального робота. Тестирование обычно проводится с помощью компьютерных симуляций, которые являются эффективным и недорогим способом убедиться в работоспособности новых методов, корректности алгоритмов и возможности их применения в существующих робототехнических системах [I. Afanasyev et al., 2015]. Лучший способ создать окружающую среду в симуляции – это использовать реальные сенсорные данные. Для нашего исследования SLAM применительно к гетерогенным группам роботов, использующих совместно создаваемую карту и многоагентное планирование маршрута, мы используем Robot Operating System (ROS). Однако, до сих пор не существовало простого и удобного инструмента для обработки и импорта реальных сенсорных данных, и разработка утилиты с таким функционалом является основным вкладом нашей работы.

Статья организована по следующему принципу. Глава 1 формулирует цели проекта и начальные условия симуляции. В главе 2 дан обзор фильтров и дано обоснование выбора для реализации. Глава 3 содержит в себе детали процесса импорта карты в симуляцию. Глава 4 демонстрирует дополнительные возможности утилиты. Далее подводятся итоги работы.

## **1. Цель и настройки системы**

### **1.1 Цель исследования**

Основная цель нашего исследования связана с изучением взаимодействия гетерогенных групп роботов, состоящих из наземных и малых воздушных автономных устройств с особым вниманием к работе в изменчивой среде. Окружающая среда может быть представлена в виде неточной, неполной или неактуальной картой. Для обеспечения надежной автономной навигации роботы совместно изучают окружающую среду и планируют действия, принимая в расчет различные источники

неопределенности. Мы достигли прогресса в SLAM и планировании пути в совместно создаваемой карте [V. Indelman et al., 2015], и продолжим изучать подходы, которые позволят учитывать изменчивость среды. В частности, такой фреймворк позволит роботам правильно действовать, улучшая локализацию и качество создаваемой карты даже в условиях недостаточности GPS сигнала.

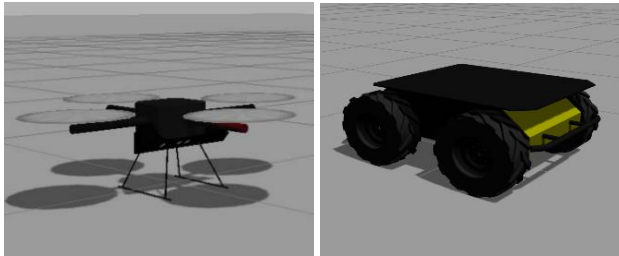


Рис. 1. Квадрокоптеры стандартной конструкции с сенсором Kinect (слева), модель робота Husky в Gazebo (справа)

Наша первая и очень упрощенная модель взаимодействия состояла из двух квадрокоптеров, строивших карту на основе данных с сенсоров Kinect, и робота Husky, планировавшего свой маршрут [M. Sokolov et al., 2016] (Рис. 1). Husky использовал данные с квадрокоптеров и свой лидар для создания графа Вороного, строил основной маршрут и следовал ему, внося локальные корректировки для избегания столкновений с препятствиями. Симуляция проводилась в искусственной среде ROS/Gazebo и все фазы работы роботов (3D-картографирование, вычисление пути и передвижение соответственно) очень сильно зависели от изменений окружающего ландшафта. Чтобы усовершенствовать симуляцию, для создания ландшафта мы хотим использовать карту, основанную на реальных сенсорных данных.

## 1.2 Условия симуляции

В нашем исследовании мы используем ROS Indigo (Robot Operating System) вместе с симулятором Gazebo 2.2, в котором имеется встроенный физический движок, удобные программные и графический интерфейсы, что позволяет создавать высококачественные симуляции. Мы осуществляем картографирование, используя модели решеток занятости и октокарт в средах ROS/Gazebo.

Решетка занятости является сеткой значений, каждое из которых обозначает наличие препятствия в указанной области [Open Source Robotics

Foundation, 2017]. Значения могут быть бинарными (0 для пустых ячеек, 1 для занятых препятствием ячеек) или принимать значения в заданном диапазоне, обозначая неровности поверхности и проходимость указанной области [S. Singh et al., 2000]. Когда значения занятости варьируются от 0 до 255, решетка занятости может быть визуализирована в виде изображения в оттенках серого с значениями равными 0 и 255 для полностью свободных и абсолютно непроходимых областей соответственно (рис. 3).

Другим способом хранить данные о занятости 3D-пространства является использование октокарт. Для эффективного хранения информации весь объем пространства делится на воксели (англ. voxels) и состояние каждого вокселя сохраняется в узлах соответствующего октодерева (дерева с коэффициентом ветвления равным 8) [Hornung et al., 2013]. Это позволяет управлять разрешением карты путем ограничения глубины октодерева. Пример деления объема на воксели и соответствующее делению октодерево показаны на рис. 2.

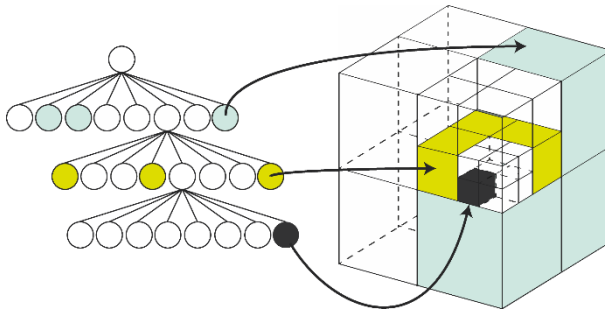


Рис. 2. Пример деления объема на воксели и соответствующее делению октодерево

### 1.3 Входные данные на основе реальных сенсорных данных

Чтобы создать реалистичную симуляцию и удостовериться в применимости предложенного решения мы использовали карту реального помещения. Данные были получены в результате лазерного сканирования в Лаборатории Автономного движения и Восприятия в Технионе – Израильском Институте Технологии (рис.3, слева). Вторая карта (рис.3, справа) была получена с сайта Hackaday (<https://hackaday.com/tag/lidar/>). Карты были созданы с помощью ROS-пакета *gmapping* и могут быть сохранены с помощью ROS-пакета *map\_server*. Пиксели могут принимать три возможных значения: «черный» для занятых областей, «белый» для пустого пространства и «серый» для тех областей, для которых данные

сканирования отсутствуют. Для поддержки функционала планирования маршрута карты должна быть отфильтрованы перед импортом в Gazebo.

## 2. Фильтрация карты

Существующие фильтры для уменьшения шумов на изображениях можно с определенной долей условности разделить на следующие типы: линейные, нелинейные и фильтры с нечеткой логикой [NachtegaeI, M, et al., 2001]; отдельно стоит упомянуть salt-and-pepper фильтры [Toh et al., 2010].

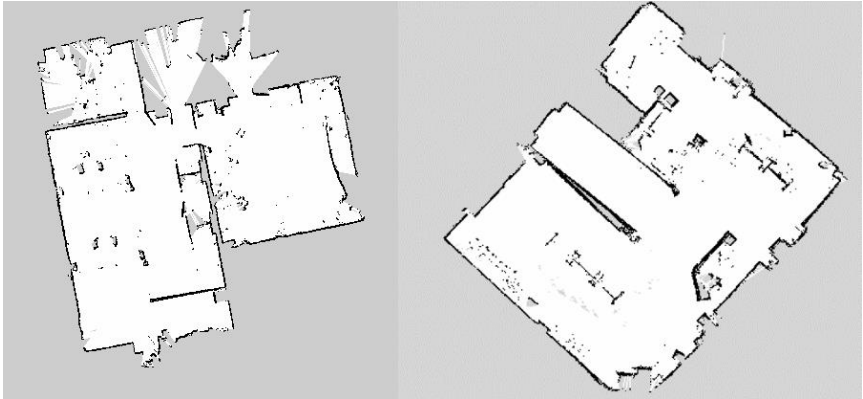


Рис. 3. Исходные карты на основе сенсорных данных

### 2.1 Выбор фильтра для реализации

На исходной карте пиксели помечены как «белые» (свободные области), «черные» (препятствия) или «серые» (области, не попавшие в зону обзора сенсоров). Это означает, что зашумленные пиксели имеют точно такие же значения, что и незашумленные. Более того, все шумы являются импульсными, т.е. они не равномерно распределены по карте.

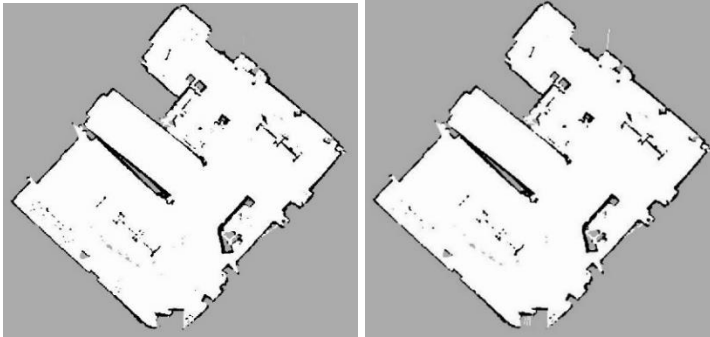


Рис. 4. Исходная карта на основе сенсорных данных, отфильтрованная стандартным (слева) и модифицированным (справа) медианным фильтром

Наличие только трех возможных значений пикселей делает невозможным использование всех фильтров, которые при фильтрации генерируют средние значения, т.к. эти значения не могут быть интерпретированы в терминах занятости или незанятости соответствующей области на карте. Фильтры с нечеткой логикой потенциально могут показать себя лучше, но их применение требует значительных усилий для адаптации алгоритмов к конкретной карте. Salt-and-pepper фильтры могут показаться лучшим выбором для устранения импульсных шумов, но их слабость – это этап определения зашумленных пикселей, т.к. в нашем случае невозможно отличить зашумленный пиксель от незашумленных в силу отсутствия разницы в их значениях.

Поэтому мы используем нелинейный медианный фильтр, который не генерирует средние значения, эффективно устраняет импульсные шумы и прост в реализации. Фильтр реализован на языке C++ и оформлен в виде ROS-пакета. На вход принимается решетка занятости, на выход же подается отфильтрованная решетка занятости в том же формате. На рис. 4 показана схема взаимодействия фильтра с другими ROS-узлами. Все фазы – обработка исходной карты, ее фильтрация и сохранение были объединены в одну утилиту со множеством настраиваемых параметров. Такое объединение делает фильтрацию легким и удобным. Пример отфильтрованной карты показан на рис. 4 (слева) и на рис. 5 (посередине).

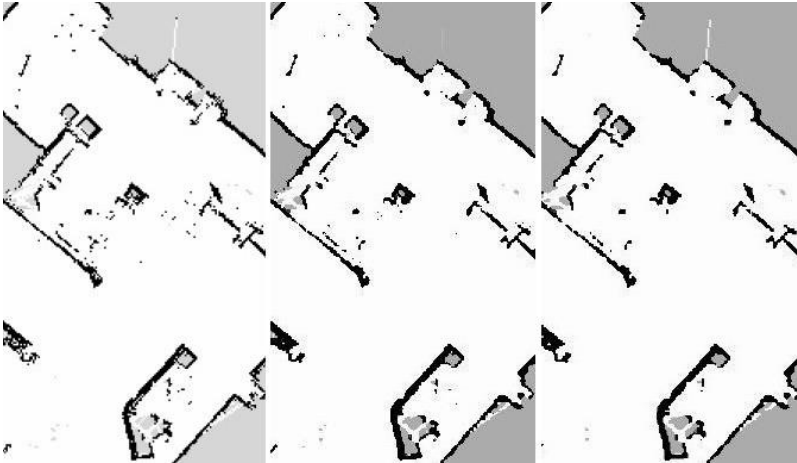


Рис. 5. Увеличенные области исходной карты (слева), отфильтрованной медианным фильтром (в центре) и отфильтрованной модифицированной медианным фильтром (справа)

Однако мы заметили, что тонкие структуры вроде внутренних перегородок исчезли в ходе фильтрации. Чтобы устранить этот недостаток, мы модифицировали медианный фильтр. Его улучшенная версия не воздействует на пиксели, ближайшие соседи которых по строке или столбцу имеют то же значение, что и сам фильтруемый пиксель. Эти соседи в принятой нами нотации будут обозначаться как  $(i, j-1)$ ,  $(i, j+1)$  в случае строки и  $(i-1, j)$ ,  $(i+1, j)$  в случае столбца. Результат работы модифицированного фильтра показан на рис. 4 (справа) и на рис. 5 (справа).

### 3. Импорт карты

Мы создали ландшафт, используя SDF-элемент *heightmap*, которому требуется файл изображения в качестве базы для создания рельефа. Также этот элемент позволяет легко контролировать размеры и максимальную высоту создаваемой карты высот. Попытка использовать исходную карту без обработки была неудачной, т.к. высоты в созданном ландшафте были инвертированы (на левом изображении рис. 6 можно увидеть данный эффект). Такое поведение симулятора вызвано требованиями к базовому изображению элемента *heightmap*: изображение должно быть формате PNG и в режиме оттенков серого, вместо RGB-режима, использующегося по умолчанию.

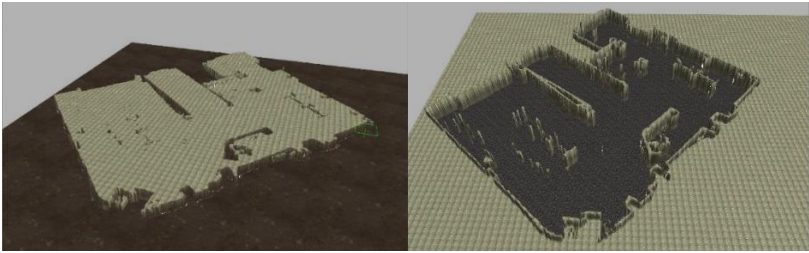


Рис. 6. Результат импорта изображения в RGB-режиме: инвертированные высоты (слева) и успешно импортированная карта высот (справа)

Наша утилита решает эту проблему, проводя все необходимые преобразования в автоматическом режиме. Преимуществом здесь является возможность использовать в качестве базового изображение любого формата. На выходе утилита предоставляет отфильтрованную, отредактированную и полностью готовую к использованию в симуляции карту. Карта высот на основе такого изображения поддерживает обработку столкновений, наложение текстур, обработку света и теней. На правом изображении рис. 6 демонстрируется успешно созданная окружающая среда на основе исходной карты (рис. 3 справа).

### 3.1 Проверка работоспособности симуляции

Чтобы проверить работоспособность и качество созданного утилитой ландшафта, мы провели решение навигационных задач с упомянутыми выше роботами Husky и двумя квадрокоптерами. Во время тестирования производительность симуляции отслеживалась с помощью показателя Real Time Factor (RTF), который отображает отношение времени, прошедшего в симуляции ко времени, которое понадобилось для обработки этого периода времени. Например, если обработка 1 секунды работы симуляции занимает 4 секунды реального времени, то RTF равен 0.25. Таким образом, чем выше RTF, тем эффективнее симуляция.

Несколько моделей роботов были использованы для проверки. Для двух передвигающихся квадрокоптеров RTF равнялся 0.7. Для двух квадрокоптеров с роботом Husky показатель RTF падал до неприемлемого значения 0.01; при использовании с двумя квадрокоптерами робота TurtleBot [M. W. Tully Foote, 2017] значение RTF снижалось до приемлемого для комфортной работы 0.3. Это подтвердило наше предположение о сложной обработке взаимодействия между картой высот и роботом Husky. На рис. 7 показана симуляция с роботом TurtleBot и двумя квадрокоптерами, выполняющими навигационные задачи на созданном по отфильтрованной карте (рис. 4 правое изображение) ландшафте.



Рис. 7. Финальная карта с роботами TurtleBot и двумя квадрокоптерами, выполняющими навигационные задачи

#### 4. Возможности утилиты

После запуска исходная карта обрезается (или расширяется), ее режим отображения меняется на режим оттенков серого. Далее идут этапы фильтрации и инверсии цветов, если они указаны пользователем. После этого карта сохраняется в формате PNG. Последним шагом является создание world-файла для Gazebo и запуск симулятора. Для удобства вся утилита интегрирована с ROS.

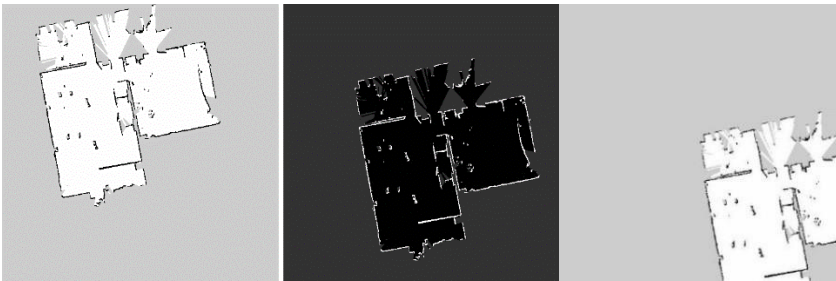


Рис. 8. Результаты запусков утилиты с разными параметрами



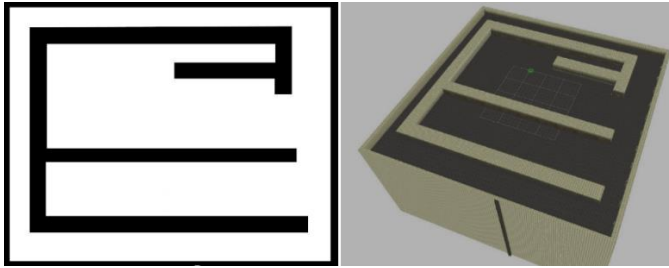


Рис. 9. Пример работы утилиты – из исходного изображения (слева) в среду Gazebo (справа)

## Заключение

В этой статье мы представили разработку утилиты для автоматического создания реалистичных ландшафтов в среде симулятора Gazebo на основе изображений (рис. 9) и карт, созданных в ходе сенсорных исследований реального мира. Утилита позволяет проводить фильтрацию и полную подготовку изображения к импорту в качестве карты высот в Gazebo в автоматическом режиме.

В дальнейшем мы планируем проверить работоспособность представленной утилиты, используя модели роботов, имеющиеся в ROS, в том числе и модель робота Инженер [M. Sokolov et al., 2016]. Утилита будет дорабатываться и будет в свободном доступе в качестве ROS-пакета.

## Список литературы

- [Fuentes-Pacheco et al., 2015] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, November 2015.
- [A. Buyval et al., 2016] A. Buyval, I. Afanasyev, and E. Magid, Comparative analysis of ROS-based Monocular SLAM methods for indoor navigation, *The 9th Int. Conf. on Machine Vision*, 2016.
- [R. W. Wolcott et al., 2014] R. W. Wolcott, R. M. Eustice. *Visual Localization within LIDAR Maps for Automated Urban Driving*, 2014.
- [I. Afanasyev et al., 2015] I. Afanasyev, A. Sagitov, and E. Magid. ROS-Based SLAM for a Gazebo-Simulated Mobile Robot in Image-Based 3D Model of Indoor Environment. *Advanced Concepts for Intelligent Vision Systems*, Springer International Publishing, pp.273-283, 2015.
- [V. Indelman et al., 2015] V. Indelman, L. Carlone and F. Dellaert, "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments.", *The Int. Journal of Robotics Research*, vol. 34, no. 7, pp. 849-882, 2015.

- [**M. Sokolov et al., 2016**] M. Sokolov, R. Lavrenov, A. Gabdullin, I. Afanasyev and E. Magid. 3D modelling and simulation of a crawler robot in ROS/Gazebo. The 4th Int. Conf. on Control, Mechatronics and Automation, 2016.
- [**Open Source Robotics Foundation, 2017**] Open Source Robotics Foundation, "nav\_msgs/OccupancyGrid Documentation"
- [**S. Singh et al., 2000**] S. Singh, R. Simmons, T. Smith, A. Stentz, V. Verma, A. Yahja & K. Schwehr. Recent Progress in Local and Global Traversability for Planetary Rovers. IEEE Int. Conf. on Robotics and Automation, 2000.
- [**Hornung et al., 2013**] Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3), 189-206.
- [**S. K. Johannes Meyer, 2014**] S. K. Johannes Meyer, "hector\_quadrotor - ROS Wiki," [http://wiki.ros.org/hector\\_quadrotor](http://wiki.ros.org/hector_quadrotor), 2014.
- [**Clearpath Robotics, 2015**] Clearpath Robotics, "Robots/Husky - ROS Wiki," <http://wiki.ros.org/Robots/Husky>, 2015
- [**Nachtegaele, M, et al., 2001**] Nachtegaele, M., Van der Weken, D., Van De Ville, D., Kerre, E., Philips, W., & Lemahieu, I. (2001). An overview of classical and fuzzy-classical filters for noise reduction. *The 10th IEEE Int. Conf. on Fuzzy Systems* (Vol. 1, pp. 3-6), 2001.
- [**Toh et al., 2010**] Toh, Kenny Kal Vin, and Nor Ashidi Mat Isa. "Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction." *IEEE signal processing letters* 17.3 (2010): 281-284.
- [**Open Source Robotics Foundation, 2014**] Open Source Robotics Foundation, "SDF," <http://sdformat.org/>
- [**T. Hearn, 2016**] T. Hearn, "GitHub - thearn/stl\_tools: Python code to produce STL geometry files from plain text, LaTeX code, and 2D numerical arrays (matrices)," [https://github.com/thearn/stl\\_tools](https://github.com/thearn/stl_tools)
- [**M. W. Tully Foote, 2017**] M. W. Tully Foote, "Robots/TurtleBot - ROS Wiki," <http://wiki.ros.org/Robots/TurtleBot>

УДК 001.57, 62-519

## МОДЕЛИРОВАНИЕ ТРЕХОСНОГО ПОДВЕСА КАМЕРЫ КВАДКОПТЕРА DJI И РАДИОУПРАВЛЕНИЯ В СРЕДЕ ROS / GAZEBO

Сагитов А. Г. (*sagitov@it.kfu.ru*)

Герасимов Ю.А. (*gerasimov@it.kfu.ru*)

Магид Е.А. (*magid@it.kfu.ru*)

Высшая Школа Информационных Технологий и  
Информационных Систем, Казанский (Приволжский)  
Федеральный Университет, Казань

**Аннотация.** В последние годы БЛА, такие как DJI Phantom 4, начали успешно использоваться в исследовательских и коммерческих приложениях. Несмотря на значительный прогресс в разработке алгоритмов управления, тестирование БЛА связано с риском повреждения относительно дорогостоящего оборудования. Для решения этой проблемы в симуляторе Gazebo доступны системы для моделирования квадоторных БЛА. Однако, существующие симуляции сильно упрощены и не моделируют коммерчески доступные БЛА. В этой статье мы представляем улучшенные пакеты моделирования квадоторного БЛА DJI Phantom, добавляя подвес и управление передатчиком RC.<sup>1</sup>

**Ключевые слова:** БЛА, радиоуправление, моделирование, Gazebo, ROS.

### Введение

Беспилотные летательные аппараты (БЛА) - летательные роботы, которые управляются дистанционно или автономно встроенной системой управления. Использование беспилотных летательных аппаратов для гражданских целей неуклонно растет, и по мере развития технологии БЛА, мы ожидаем, что их использование скоро станет стандартной практикой для решения большого количества различных задач.

---

<sup>1</sup> Работа поддержана Российским фондом фундаментальных исследований (РФФИ) и Министерством науки и технологий и Государством Израиль (совместный проект ID 15-57-06010). Часть работ выполнена в соответствии с Государственной программой конкурентоспособности Российской Федерации при Казанском федеральном университете.

Непрерывное улучшение и миниатюризация камер позволили использовать готовое оборудование, объемом менее восьми кубических сантиметров, при этом весом менее 30 граммов и цене менее ста долларов США. Также стали доступны небольшие радиоуправляемые передатчики для управления БЛА, работающие на частоте 2,4 ГГц и обеспечивающие диапазоны управления более 10 км в условиях открытой местности. Используя такое оборудование, многие типы БЛА доступны в качестве платформы для решения задач сбора данных или локализации целей в реальном мире.

Тестирование прототипов БЛА по-прежнему включает в себя риски из-за возможных сбоев, приводящих к повреждению аппаратных компонентов вплоть до полной потери БЛА. Эти проблемы делают комплексное моделирование и имитацию БЛА приоритетным направлением. В целях уменьшения стоимости и риска тестирования системы при разработке системы управления полетом БЛА, система имитации полета с использованием подходов виртуального прототипа была рассмотрена с использованием инструмента Real-Time Workshop (Matlab) и операционной системы реального времени VxWorks [Kaidi, 2016]. Моделирование квадоторных БЛА обычно выполняется с использованием коммерческих инструментов, например, Matlab/Simulink [Lu, 2011]. Для моделирования некоторыми научными группами используется игра X-plane 10 компании Laminar Research [Cho, 2015],[Nugroho, 2014].

Хотя эти инструменты обеспечивают общую достоверную симуляцию, они все же требуют значительных усилий, необходимых для тестирования высокоуровневого управления с использованием датчиков. Кроме того, коммерческие симуляторы обычно имеют ограниченные возможности модификации и улучшения. Из-за этих ограничений открытая исходная структура ROS-Gazebo обеспечивает стандартное моделирование с использованием модульной архитектуры в исследованиях робототехники и облегчает модификацию симуляций. Одно из наиболее полных имитаций полета в ROS Gazebo принадлежит исследовательской группе из TU Darmstadt [Koenig, 2004]. Основным направлением являлось включение в моделирование внешних возмущений, таких как влияние ветра на динамику БЛА. Последующие работы велись по моделированию посадки БЛА на динамические платформы используя физическую модель БЛА Мейера [Meyer, 2012].

Другой способ - моделирование аппаратного цикла (Hardware-in-the-loop, HIL), используемое для тестирования без необходимости реальных экспериментов с роботом [Odelga, 2015]. Использование моделирования HIL для тестирования платформы в режиме реального времени и реальной среды оказалось надежным подходом для тестирования критических свойств безопасности [Chandhrasekaran, 2010]. Была представлена модель

вертолета БЛА с использованием HIL, который показал экономическую эффективность для разработки общей эффективной системы управления и безопасности в [Cai, 2008].

В последние годы фреймворк ROS стал де-факто стандартом для многих связанных с робототехникой научных групп, не исключая нашу группу. Единственным недостатком симулятора ROS / Gazebo являются высокие требования к ресурсам CPU и графических карт. Альтернативой Gazebo для 3D-моделирования является программное обеспечение виртуальной экспериментальной платформы для робототехники V-REP, которое можно совместить с использованием ROS [Olivares-Mendez, 2012].

В рамках нашего научно-исследовательского проекта по взаимодействию гусеничного беспилотного наземного робота «Сервосила Инженер» и группы БЛА предполагается задействовать квадрокоптеры DJI Phantom 4, и нами были рассмотрены совместимые с этой моделью симуляторы. Хотя существует официальный симулятор DJI (DJI GO), он является закрытым продуктом и имеет существенные ограничения. Использование пакета `hector_quadrotor` в среде Gazebo позволяет одновременно моделировать различные аспекты БЛА, такие как динамика полета, инерциальные измерительные устройства (ИИУ), камеры, сложные условия полета и т.д., и позволяет с модифицировать модель БЛА DJI Phantom для увеличения реалистичности. Пакет `hector_quadrotor` был разработан для БЛА Гектор Квадротор [Meyer, 2012], и содержит аппаратное описание URDF для квадоторного БЛА, его диспетчеров полета и файлы для запуска моделирования квадотора в Gazebo.

При использовании пакета `hector_quadrotor` мы столкнулись с несколькими проблемами. Во-первых, камера, встроенная в БЛА, закреплена на базе квадотора без подвеса, поэтому стабилизация видео отсутствует. Во-вторых, отсутствует модуль для управления моделируемым БЛА при помощи радиопередатчика.

## 1. Подвес

При движении БЛА в неблагоприятных условиях возникают вибрации, нарушающие стабильность видеопотока. Основным способом достижения стабильного видеопотока БЛА является использование карданного подвеса для камеры. Подвес - это регулируемый держатель для камеры или инструмента, предназначенный для поддержания постоянного уровня устройства, несмотря на внешние факторы, т.е. стабилизирующий и антивибрационный аппарат. Подвес стабилизирует камеру БЛА при движении и гарантирует качественный видеопоток [Rajesh, 2015]. При использовании бортовой стабилизированной камеры, БЛА может выполнять аэрофотосъемку для выполнения различных миссий, таких как,

например, отслеживание цели [Quigley, 2005] или мониторинг окружающей среды [Zhou, 2005] при помощи фотограмметрии [Eisenbeiss, 2009], что требует качественного программного обеспечения для управления подвесом камеры. Механика подвеса проста: он состоит из концентрических колец, которые поворачиваются под прямым углом вокруг камеры, что помогает поддерживать устройство во время движения БЛА.

DJI Phantom 4 имеет стабилизированный подвес камеры, представленный на рисунке 1. Двигатели на каждой из трех осей позволяют прибору оставаться неподвижным и без вибраций при движении БЛА. Если БЛА вращается по часовой стрелке, подвес автоматически поворачивает камеру на соответствующий угол против часовой стрелки, обеспечивая постоянный видеосигнал.



Рис. 1. Подвес DJI Phantom 4

На основе технической спецификации подвеса DJI, мы построили 3D-модель подвеса в формате URDF (рисунок 2) в системе автоматизированного проектирования SolidWorks. Необходимые физические параметры модели были заданы из технических характеристик реального подвеса. Использована физическая модель подвеса, описанная в работе [Родионов В. и др., 2011]. Для выполнения стабилизации, необходимо получить значения углов подвеса камеры в соответствующей системе координат. При помощи ИИУ, измеряются значения и используются расчеты для оценки позиционирования и ориентации - крена, тангажа и рыскания.

Используя приемник GPS, мы получаем данные о высоте, положении и ориентации БЛА; объединяя их с показателями ИИУ, рассчитываются отношения между системой координат БЛА и глобальной системой координат. По этим расчётам создается сигнал ШИМ для управления контроллером подвеса камеры в нужном положении, чтобы реализовать

стабилизацию камеры. Следует подчеркнуть, что данные ИИУ иногда имеют значительную погрешность, которая может повлиять на решение систем уравнения для определения ориентации и направления. Поэтому необходима начальная калибровка ИИУ, и такая возможность критична для БЛА, задачи которого предполагают сложные маневры.

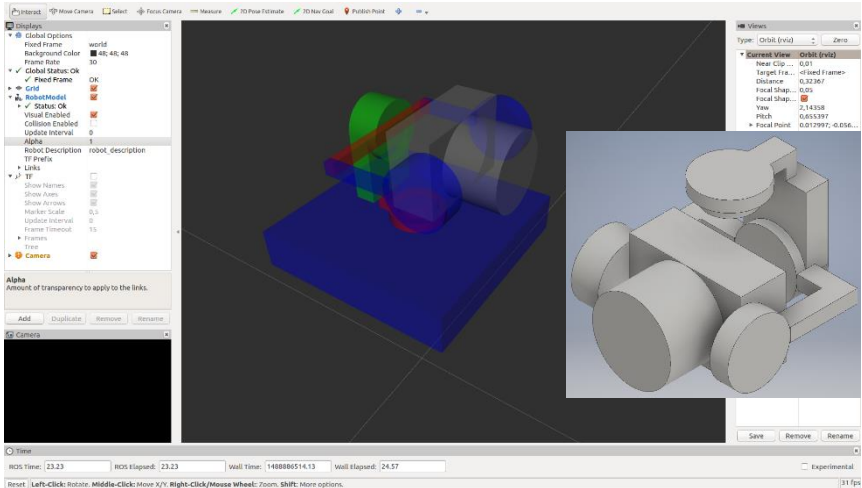


Рис. 2. Трехмерная модель подвеса

Мы модифицировали оригинальный пакет `hector_quadrotor`, включив модель каркаса и применив совместный `state_publisher` в сочетании со статическим `transformation_publisher` для построения дерева преобразований. Мы также реализовали управление подвесом, используя пакет `ros_control`.

## 2. Радиоуправление

Управление БЛА осуществляется по линии радиосвязи. Обычно радиоконтроллер состоит из радиопередатчика, управляемого (теле)оператором, и БЛА. Сформированная ими система управления с открытым контуром отправляет команду через радиопередатчик и принимает команду приемником на стороне БЛА. Передатчик отправляет данные на основе нажатых кнопок на устройстве управления оператора, которые закодированы и заданы в качестве входных данных для БЛА.

Современные RC-передатчики оснащены специальным разъемом для обучения методом ведущее-ведомое устройство (Master-Slave). Ведомый RC-передатчик (Slave) подключается к ведущему (Master) кабелем, а ведущий передатчик имеет специальный переключатель, при включении

которого управление выполняется дочерней консолью. При этом, оператор ведущего передатчика может взять ведомый передатчик под контроль в любой момент времени. Используя эту функцию, мы реализовали управление моделируемым БЛА при помощи RC-контроллера.

Мы используем 9-канальный передатчик Turnigy 9X (рисунок 3), который имеет разъем для кабеля в виде 3,5 мм контактного разъема, что позволяет получить доступ к выходному сигналу передатчика с помощью кабеля на микрофонный вход компьютера при запуске симуляции. Значения уровня каждого канала кодируется фазово-импульсной модуляцией. Формат этой модуляции можно настроить с консоли. Пример сигнала, представлен на рисунке 4.



Рис. 3. Передатчик Turnigy 9X

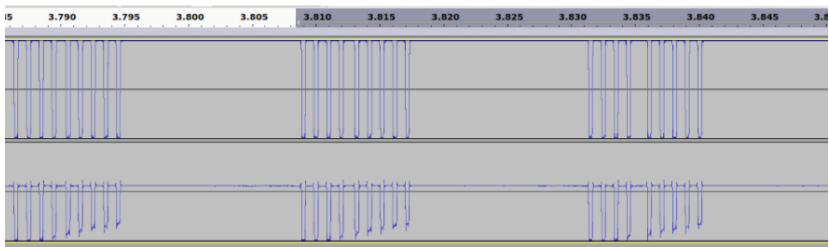


Рис. 4. Управляющий сигнал

Одним из основных осложнений использования импульсной модуляции положения является то, что приемник должен синхронизировать локальные часы с началом каждого кода. Это действие обычно выполняется как



дифференциальная импульсная модуляция, когда каждое положение импульса кодируется относительно предыдущего импульса. Он реализован таким образом, что приемник должен только измерять разницу от прибытия следующих импульсов. Он выполняется таким образом, чтобы ошибки символа влияли только на соседние, так что ошибка в измерении задержки отдельного импульса повлияет только на два импульса. Ширина канальных импульсов зависит от положения джойстика; ширина импульса между канальными импульсами составляет 300 мкс, а полный цикл - 22,5 мс.

Чтобы прочитать эти импульсы, был создан скрипт на языке Python с использованием библиотеки PyAudio (оболочка Python поверх PortAudio). Скрипт считывает информацию, поступающую от гнезда микрофона, и разделяет полученный с гнезда микрофона сигнал на временные отрезки. Последовательно проходя по отрезкам сигнала, скрипт находит ширину и логические уровни отрезков. В темы симулируемого БЛА записываются декорированные из отрезков управляющие команды. Если ширина положительного импульса больше, чем максимальная ширина импульса канала, это синхронный сигнал.

### 3. Результаты моделирования

Реализованные функции и модель были протестированы в симуляторе в Gazebo 7.0 в ROS Kinetic Kame. Полет в среде моделирования проводился несколько раз с использованием автономного и ручного управления с помощью RC-передатчика. Результаты моделирования оказались реалистичными и надежными. Подвес смог поддерживать желаемую ориентацию, в то время как БЛА контролировался контроллером RC. В аспекте надежности результаты моделирования показывают общую согласованность с поведением реального подвеса во всех тестовых случаях. Результаты показывают, что мы успешно добавили полностью функциональный подвес в пакет симуляции `hector_quadrotor`. Моделирование БЛА с подвесом представлено в рисунках 5 и 6.

### Заключение

В данной работе представлен подход к моделированию DJI Phantom 4 в среде Gazebo с использованием модифицированного пакета `hector_quadrotor`. К существующей модели был добавлен функциональный подвес для камеры и возможность управления квадатором с радиопередатчика, что позволяет проводить более реалистичную симуляцию.

В нашей будущей работе мы планируем продолжить доработку пакета `hector_quadrotor` ROS, делая его более подходящим для моделирования DJI

Phantom 4 в среде ROS / Gazebo. Планируется интегрировать новую модель БЛА в нашу предыдущую работу [Sokolov et al, 2016], чтобы иметь возможность моделировать гетерогенную группу, состоящую из гусеничного робота «Сервосила Инженер» и DJI Phantom 4 в симуляторе Gazebo.

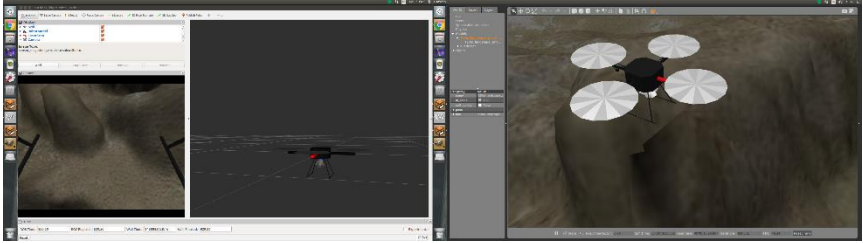


Рис. 5. БЛА в симуляторе Gazebo

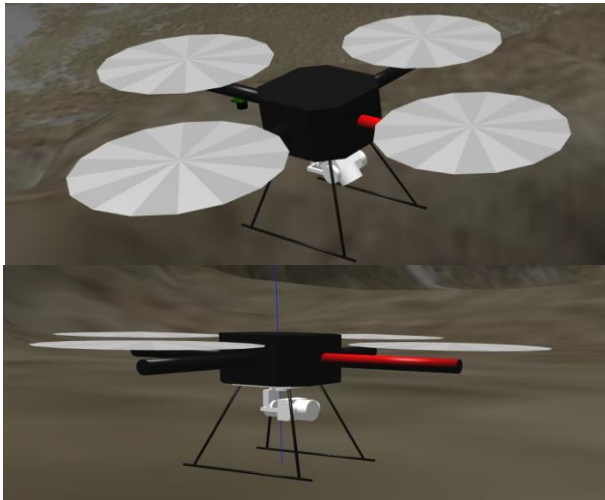


Рис. 6. БЛА в симуляторе Gazebo (крупный план)

## Список литературы

- [Kaidi, 2016] W. Kaidi, L. Chuntao, C. Peng, and F. Ying, "Design of real-time and multi-task UAV simulation system based on rapid prototyping," // IEEE Chinese Guidance, Navigation and Control Conference, Aug 2016, pp. 930–936.
- [Lu, 2011] P. Lu and Q. Geng, "Real-time simulation system for UAV based on Matlab/Simulink," IEEE 2nd Int. Conf. on Computing, Control and Industrial Engineering, Wuhan, 2011, pp. 399-404.

- [**Cho, 2015**] K. Cho, J. Shin, and T. Kuc, “Design of quadrotor controller for emergency situation using xplane,” // 12th Int. Conf. on Ubiquitous Robots and Ambient Intelligence, Oct 2015, pp. 311–314.
- [**Nugroho, 2014**] L. Nugroho, “Comparison of classical and modern landing control system for a small unmanned aerial vehicle,” // Int. Conf. on Computer, Control, Informatics and Its Applications, Oct 2014, pp. 187–192.
- [**Koenig, 2004**] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” // IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, vol. 3, Sept 2004, pp. 2149–2154 vol.3.
- [**Olivares-Mendez, 2012**] M. A. Olivares-Mendez, P. Campoy, I. Mellado-Bataller, and L. Mejias, “See-and-avoid quadcopter using fuzzy control optimized by cross-entropy,” // IEEE Int. Conf. on Fuzzy Systems, June 2012, pp. 1–7.
- [**Odelga, 2015**] M. Odelga, P. Stegagno, H. H. Blthoff, and A. Ahmad, “A setup for multi-uav hardware-in-the-loop simulations,” // Workshop on Research, Education and Development of Unmanned Aerial Systems, Nov 2015, pp. 204–210.
- [**Chandhrasekaran, 2010**] V. K. Chandhrasekaran and E. Choi, “Fault tolerance system for uav using hardware in the loop simulation,” // 4th Int. Conf. on New Trends in Information Science and Service Science, May 2010, pp. 293–300.
- [**Cai, 2008**] G. Cai, B. M. Chen, T. H. Lee, and M. Dong, “Design and implementation of a hardware-in-the-loop simulation system for small-scale UAV helicopters,” // 2008 IEEE Int. Conf. on Automation and Logistics, Sept 2008, pp. 29–34.
- [**Meyer, 2012**] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk, “Comprehensive simulation of quadrotor UAVs using ROS and gazebo,” // Simulation, Modeling, and Programming for Autonomous Robots. Springer Berlin Heidelberg, 2012, pp. 400–411. [Online]. Available: [https://doi.org/10.1007%2F978-3-642-34327-8\\_36](https://doi.org/10.1007%2F978-3-642-34327-8_36)
- [**Rajesh, 2015**] R. J. Rajesh and P. Kavitha, “Camera gimbal stabilization using conventional PID controller and evolutionary algorithms,” // Int. Conf. on Computer, Communication and Control, Sept 2015, pp. 1–6.
- [**Quigley, 2005**] M. Quigley, M. A. Goodrich, S. Griffiths, A. Eldredge, and R. W. Beard, “Target acquisition, localization, and surveillance using a fixed-wing mini-uav and gimbaled camera,” // Proc. IEEE Int. Conf. on Robotics and Automation, April 2005, pp. 2600–2605.
- [**Zhou, 2005**] G. Zhou, C. Li, and P. Cheng, “Unmanned aerial vehicle (uav) real-time video registration for forest fire monitoring,” // Proc. IEEE Int. Geoscience and Remote Sensing Symposium, vol. 3, July 2005, pp. 1803–1806.
- [**Eisenbeiss, 2009**] H. Eisenbeiss, UAV photogrammetry. // ETH, 2009.
- [**Sokolov, 2016**] M. Sokolov, R. Lavrenov, A. Gabdullin, I. Afanasyev, and E. Magid, “3d modelling and simulation of a crawler robot in ros/gazebo,” // Proc. 4th Int. Conf. on Control, Mechatronics and Automation, ser. ICCMA '16. New York, NY, USA: ACM, 2016, pp. 61–65.
- [**Родионов В. и др., 2011**] Родионов В. И., Михед А. Д. Динамика системы стабилизации и наведения линии визирования с двухосным наклонным кардановым подвесом // Известия ТулГУ. Технические науки. 2011. [Online]. Available: <http://doi.acm.org/10.1145/3029610.3029641>

УДК 007.52

## ПОИСК МАРШРУТА ДЛЯ НАЗЕМНОГО РОБОТА: МОДИФИЦИРОВАННЫЙ АЛГОРИТМ ПЛАНИРОВАНИЯ НА ОСНОВЕ СПЛАЙНОВ

Р.О. Лавренев (*lavrenov@it.kfu.ru*)Е.А. Магид (*magid@it.kfu.ru*)Высшая Школа Информационных Технологий и Систем,  
Казанский Федеральный Университет, Казань

**Аннотация.** Методы, основанные на принципе потенциальных полей, играют важную роль в задачах планирования пути. Но они страдают от проблемы застревания пути в локальных минимумах функции потенциального поля. В этой статье мы предлагаем модификацию оригинального сплайн-алгоритма для планирования маршрута наземного мобильного робота путем интеграции графа Вороного, благодаря чему успешно решаем проблему локальных минимумов. В новый алгоритм добавлены дополнительные критерии видимости стартовой и целевой точек для оптимизации выбора пути. Алгоритм был реализован в среде Matlab, и результаты моделирования продемонстрировали, что нам удалось преодолеть проблемы исходного алгоритма.<sup>1</sup>

**Ключевые слова:** планирование маршрута; мобильный робот; граф Вороного; метод потенциального поля; безопасность маршрута.

### Введение

В настоящее время разработки в сфере робототехники направлены на замену человеческого труда в случаях, когда это существенно ускорит или обезопасит какой-либо производственный процесс или проведение поисково-спасательных операций в опасных для людей условиях [Magid и др., 2011]. При этом, современные мобильные роботы должны обладать такими критическими навыками как взаимодействие с пользователями

---

<sup>1</sup> Работа поддержана Российским фондом фундаментальных исследований (РФФИ) и Министерством науки и технологий и Государством Израиль (совместный проект ID 15-57-06010). Часть работ выполнена в соответствии с Государственной программой конкурентоспособности Российской Федерации при Казанском федеральном университете.

[Pipe и др., 2014] и другими роботами [Rosenfeld и др., 2015] [Ronzhin и др., 2016] [Panov и др., 2017], автономное перемещение [Magid, 2006], выстраивание карты окружающего пространства и локализации на карте (SLAM) [Buyval и др., 2016]. Все вышеперечисленные навыки в обязательном порядке требуют интеграции с планировщиком маршрута.

Классический, но остающийся популярным, метод планирования маршрута – метод потенциального поля [Khatib и др., 2016]. В этом подходе со стороны целевой точки на робота действуют силы притяжения, а со стороны препятствий – силы отталкивания. Робот движется по линии градиента в направлении минимума потенциального поля. Потенциалы поля могут рассчитываться как для всей карты целиком, когда она известна заранее (глобальное планирование пути), так и только для некоторой локальной области карты, когда карта неизвестна роботу (локальное планирование пути). Недостатками этого метода является застревание робота в местах локальных минимумов потенциального поля или возникающие в узких проходах между препятствиями осцилляции.

В предыдущем исследовании нами был предложен метод планирования пути для мобильного робота, основанный на сплайнах [Magid и др., 2006]. Данный алгоритм использует принципы методов потенциальных полей и критерии оптимизации маршрута - такие как длина, кривизна и безопасность маршрута. Чтобы улучшить алгоритм, добавить гибкость для оптимизации пути и возможность быстрого динамического перепланирования, мы интегрируем подход диаграммы Вороного и новый критерии видимости точек старта и цели в наш исходный метод.

## **1. Оригинальный метод планирования пути**

Алгоритм, описанный в работе [Magid и др., 2006], предлагает решение задачи планирования пути для мобильного робота, представленного точкой в пространстве конфигураций [Choset, 2005], в известной среде со статическими препятствиями в виде конечной группы пересекающихся кругов разного размера. Идея такого представления в том, что любой многоугольник можно аппроксимировать группой кругов.

### **1.1 Целевая функция**

Чтобы гарантировать построение пути, не пересекающего препятствия, отталкивающая потенциальная функция должна обладать высоким значением внутри препятствия и на его границе, и небольшим значением вне препятствия. Во время локальной процедуры оптимизации пути, высокое значение потенциальной функции внутри препятствия выталкивает все точки пути из него наружу, чтобы минимизировать стоимость пути. Определим положение робота в момент времени  $t$  как

конфигурацию  $q(t)=(x(t),y(t))$ . Тогда вклад одного круга (где круг является частью препятствия) в общую функцию потенциала отталкивания определяется следующим уравнением:

$$U_{rep}(q) = 1 + \tanh(\alpha(\rho - \sqrt{(x(t) - x)^2 + (y(t) - y)^2})), \quad (1)$$

где  $\rho$  - это радиус круга (препятствия) с центром в точке с координатами  $(x,y)$ , а  $\alpha$  - это эмпирически определяемый параметр, который отвечает за выталкивание пути за границы препятствия. Рисунок 1 (справа) демонстрирует пример выбора параметра  $\alpha=0.5$  для среды с одним препятствием, которое образовано тремя пересекающимися кругами (слева). Потенциальная функция имеет четко выделяющиеся пики на пересечениях кругов.

**Топология**  $T(q)$  – функция, учитывающая все  $N$  кругов окружающей среды и их влияние на работа на протяжении всего пути, которая определяется параметрическим уравнением на интервале  $[0,1]$ :

$$T(q) = \sum_{j=0}^{N-1} \int_{t=0}^1 U^j_{rep}(q) \cdot \delta l(t) \cdot dt, \quad (2)$$

где  $\delta l(t)$  длина отрезка:

$$\delta l(t) = \sqrt{(x'(t))^2 + (y'(t))^2} \quad (3)$$

Функция **кривизны** пути также определяется параметрическим уравнением:

$$R(q) = \sqrt{\int_{t=0}^1 (x''(t))^2 + (y''(t))^2 dt} \quad (4)$$

Функция **длины** пути, суммирует длины всех сегментов пути:

$$L(q) = \int_{t=0}^1 \delta l(t) \cdot dt \quad (5)$$

Таким образом, итоговая стоимость пути состоит из суммы трех компонентов:

$$F(q) = \gamma_1 T(q) + \gamma_2 R(q) + \gamma_3 L(q), \quad (6)$$

где  $\gamma_{i=1,3}$  - числовые значения влияния каждого компонента на общую стоимость маршрута. В частности, значение влияния потенциального поля  $\gamma_1 = \beta/2$ , где параметр  $\beta$  коррелирует с  $\alpha$  из уравнения (1) [Magid, 2006]. В программной реализации интегралы считаются из разбиения кривой на отрезки с длиной, стремящейся к нулю.

## 1.2 Алгоритм

Оригинальный сплайн-алгоритм является итеративным. Путь на первой итерации представляет из себя сплайн, который определяется тремя точками:  $S$ ,  $T$  и равноудаленной точкой, лежащей на прямом отрезке, соединяющем точки  $S$  и  $T$ . Дальнейшие итерации расчета пути используют

уравнение (6) в роли целевой функции стоимости и оптимизируют маршрут относительно его стоимости при помощи метода Нелдера-Мида [Lagarias и др., 1998] (симплекс-метод). Лучший результат (т.е. маршрут) каждой итерации служит начальным значением (маршрута) для следующей итерации. Маршрут перестраивается на каждой итерации, используя информацию с предыдущего этапа, увеличивая количество точек сплайна на единицу и корректируя параметры целевой функции стоимости. Алгоритм завершается, если количество итераций превышает заданный пользователем предел, или если новая итерация не улучшает предыдущую.

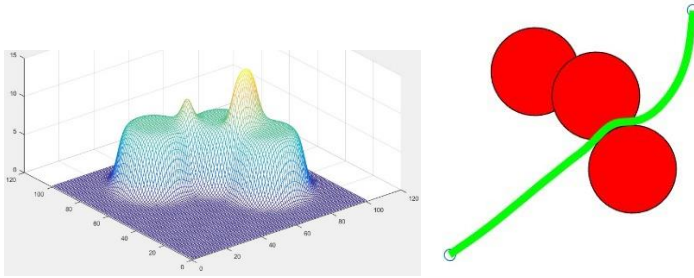


Рис. 1. Потенциальная функция одиночного препятствие, состоящего из трех кругов (слева) и путь, рассчитанный оригинальным алгоритмом (справа)

### 1.3 Слабые стороны оригинального алгоритма

Исходный сплайн-метод позволяет получить гладкий путь без столкновений с препятствиями, если каждое препятствие аппроксимируется одним кругом или линия начальной итерации не проходит через пересечение кругов. Однако, когда препятствия аппроксимированы группой пересекающихся кругов, пересечения образуют локальные максимумы потенциального поля (рис. 1, слева). Когда в такой среде начальный сплайн проходит через пересечение нескольких кругов, то целевая функция  $F(q)$  выталкивает сплайн из области пересечения и далее сплайн «застревает» в локальных минимумах, а следующий итерационный сплайн не может «перепрыгнуть» некоторые компоненты препятствия из-за локального характера процесса оптимизации. На рис. 1 (слева) демонстрируется пример с одним сложным препятствием. Локальная оптимизация успешно устраняет локальные максимумы, но даже после 18 итераций, продолжавшихся в течение 23 минут, результирующий путь все еще пересекает препятствие и, следовательно, бесполезен для движения робота (рис. 1, справа).

## 2. Интеграция дополнительного критерия оптимизации

В этом разделе мы вводим два новых критерия оценки качества маршрута – время видимости стартовой и целевой точки – и исследуем их влияние на результирующий путь. Время видимости точки – это процентное отношение длины пути, где робот сохраняет точку старта (или целевую точку соответственно) в пределах своей прямой видимости.

Эти два критерия важны, если роботу необходимо максимизировать время прямого и непрерывного визуального или радиоконтакта с контрольным устройством или маршрутизатором, который поддерживает планирование маршрута, локализацию или любые другие функции робота. Критерий учитывает отношение видимого и невидимого от стартовой (или целевой) точки участков пути. Таким образом, мы минимизируем время, которое фактически измеряется как сумма длины участков пути (предполагается постоянная скорость робота, а в будущей работе мы также распространяем это на случаи различной скорости вдоль пути), на которых контрольные точки недоступны для прямой видимости робот. Значение критерия для начальной точки старта  $S$  вычисляется по формуле:

$$I_S = 1 - \lim_{\delta, \delta(t) \rightarrow 0} \frac{\sum_{t=0}^u \text{dist}(q(t), q(t+\delta t))}{\int_{t=0}^1 \delta(t) \cdot dt} \quad (7)$$

$$\forall t \in [0, u + \delta]: [q(t), S] \cap (\cup_{j=1}^N \text{Obs}_j) = 0$$

где числитель дроби в уравнении 7 отражает длину пути, видимого из начального положения  $S$ , а знаменатель дроби отражает полную длину пути из уравнения 5 и считается через аппроксимацию кривой отрезками с длиной, стремящейся к нулю.  $q(t)$  - положение робота в момент времени  $t$ . Короткие отрезки пути, которые были пройдены между моментами времени  $t$  и  $t+\delta t$ , обозначаются  $\text{dist}(q(t), q(t+\delta t))$ . Нижняя строка уравнения 7 описывает свойство видимости: прямой отрезок  $[S; q(t+\delta t)]$  не пересекается ни с каким препятствием  $\text{Obs}_j$ , где  $j=1..N$ , а  $N$  – число препятствий в окружающей среде. Таким образом, последняя видимая точка пути от стартовой точки описывается с помощью  $q(u+\delta t)$ . Аналогичным образом описывается критерий, который показывает отношение сегментов пути с прямой видимостью целевой точки  $T$ :

$$I_T = 1 - \lim_{\delta, \delta(t) \rightarrow 0} \frac{\sum_{t=w}^{1-\delta} \text{dist}(q(t), q(t+\delta t))}{\int_{t=0}^1 \delta(t) \cdot dt} \quad (8)$$

$$\forall t \in [w, 1]: [q(t), T] \cap (\cup_{j=1}^N \text{Obs}_j) = 0$$



Уравнение 8 описывает первую точку  $q(w)$  сегмента пути  $[q(w), q(I)=T]$ , которая отмечает начало последнего сегмента пути, характеризующегося гарантированной прямой видимостью робота из целевой позиции  $T$ .

Целевая функция, которая объединяет все пять критериев, определяется следующим образом:

$$F(q) = \gamma_1 T(q) + \gamma_2 R(q) + \gamma_3 L(q) + \gamma_4 I_S + \gamma_5 I_T, \quad (9)$$

где  $\gamma_4$  и  $\gamma_5$  – весовые коэффициенты новых критериев в целевой функции.

На рис. 2 показано влияние двух новых критериев на найденный путь: в то время как при  $\gamma_4 = 10$  и  $\gamma_5 = 10$  эти критерии оказывают существенного влияния (рис. 2, слева), при  $\gamma_4 = 20$  и  $\gamma_5 = 20$  изменения траектории хорошо заметны (рис. 2, справа). Другие параметры установлены эмпирически как  $\gamma_1 = 1$ ,  $\gamma_2 = 1$  и  $\gamma_3 = 0,5$  для обоих случаев.

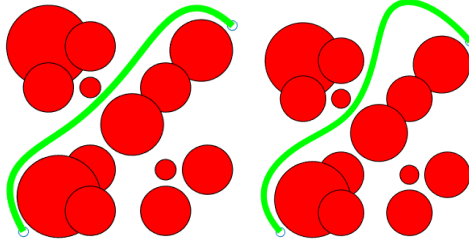


Рис. 2. Влияние критерия видимости начальной и конечной точек на маршрут

Так как оптимизация маршрута с помощью уравнения 9 выполняется только локально, влияние дополнительных параметров также локально. Путь остается в той же гомотопической группе даже в том случае, когда можно потенциально обойти какое-либо препятствие с другой его стороны. В рамках нашей будущей работы мы планируем применить оптимизацию в глобальном масштабе, которая позволит выбирать наиболее оптимальные пути среди нескольких гомотопий.

### 3. Решение на основе диаграммы Вороного

Для получения хорошего начального сплайна, мы строим граф Вороного. Для создания этого графа требуется подготовить информацию о препятствиях; процедура выполняется в два шага:

Сбор препятствий из пересекающихся кругов. Первоначально каждый круг отмечен как неиспользованный и имеет собственный индекс  $i=1...M$ , где  $M$  – число кругов в среде. Начнем с произвольного неиспользованного круга, закрепим его как принадлежащий препятствию  $O_1$  и пометим как использованный. Затем итеративно увеличиваем  $O_1$ , определяем все

неиспользованные круги, которые пересекаются с кругами препятствия  $O_1$ , закрепляем их за  $O_1$  и отмечаем как использованные до тех пор, пока в среде не останется больше неиспользованных кругов, которые пересекаются с  $O_1$ . Если остаются еще неиспользованные круги, мы произвольно выбираем другой неиспользованный круг, закрепляем его за препятствием  $O_2$  и повторяем процедуру роста. Сбор завершается, когда не остается неиспользованных кругов. Например, на рис. 2 есть 7 препятствий, которые образованы группами кругов, а на рис. 3 в среде находятся 12 препятствий. При сборке препятствий каждая пара пересекающихся кругов  $(i, j)$  образует одну точку пересечения  $\theta_{ij}$  в случае граничного касания и пару точек  $\theta_{ij}, \theta_{ji}$  в общем случае пересечения кругов.

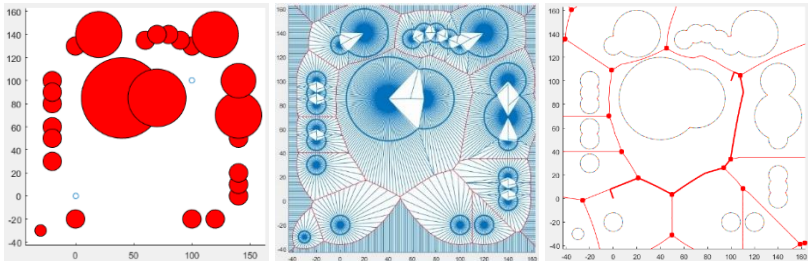


Рис. 3. Среда с препятствиями (слева), процедура построения на ней графа Вороного (центральное), построенный граф и путь в графе (справа)

Поиск внешних контуров препятствий множества  $Obst = \{O_1, \dots, O_k\}$ , где  $k$  - количество препятствий в окружающей среде. Исходя из радиуса самого маленького круга внутри препятствия  $O_1$ , границы всех кругов, принадлежащих  $O_1$ , разбиваются на короткие отрезки длиной  $\sigma$ , соединенные через точки пересечения  $\theta_{ij}, \theta_{ji}$ . Далее, каждые два сегмента соединяются, если между ними существует непрерывный путь, который построен из граничных сегментов. Эта процедура повторяется для каждого препятствия, и по ее завершению мы получаем группу контуров. Если размер последнего множества превышает  $k$ , это указывает на наличие внутренних контуров. Чтобы избавиться от таких контуров, для каждого конкретного препятствия  $O_i$  вычисляется выпуклая оболочка и удаляются те контуры, которые находятся внутри выпуклой оболочки. Таким образом, мы получаем несколько невыпуклых многоугольников, по одному для каждого препятствия из  $Obst$ , и инкапсулируем их в квадратную карту.

Далее, имея карту препятствий в виде многоугольников, строится граф Вороного [Choset, 2005]:

От границ препятствий и карты с равными промежутками строятся лучи. На рис. 3 (посередине) демонстрируется пример построения этих

лучей в виде тонких синих линий, а толстые синие линии отображают границы препятствий и карты.

Вычисляются точки пересечения лучей от разных препятствий. Точки пересечения соединены отрезками. Соединяя отрезки, мы образуем граф Вороного (тонкие красные линии на рис. 3, посередине и справа)

После построения графа Вороного ( $VG$ ), для присоединения к графу начальной позиции  $S$  и целевой позиции  $T$  на графе выбираются точки  $S'$  и  $T'$ , ближайшие к точкам  $S$  и  $T$ , соответственно. Отрезки  $[S, S']$  и  $[T, T']$  должны не пересекать препятствия. Кратчайший путь между  $S'$  и  $T'$  находится в графе  $VG$  алгоритмом Дейкстры [Dijkstra, 1959]. Так, на рис. 3 (справа) построенный путь изображен толстыми красными линиями. Любой путь между точками ( $S, T$ ) на графе Вороного  $VG$  гарантированно не пересекает препятствия и максимально безопасен в отношении расстояния от границ препятствий. Таким образом, он может обеспечить хороший начальный сплайн для исходного сплайн-метода [Magid и др., 2006].

Однако, нужно правильно выбрать точки для построения сплайна. На первом шаге выбирается точка  $S$  в качестве активной точки и находится точка  $VP_1$  на пути по графу Вороного  $VG$ , таким образом, что следующая за  $VP_1$  точка на графе  $VG$  уже не видна из  $S$  из-за окклюзии препятствиями.  $S$  добавляется к множеству  $L$ , которое хранит точки графа  $VG$ , характеризующие данный маршрут и достаточные для его описания.  $VP_1$  получает статус следующей активной точки. Затем мы снова ищем на пути самую дальнюю точку, видимую из  $VP_1$  – точку  $VP_2$ .  $VP_1$  добавляется в  $L$ ,  $VP_2$  становится следующей активной точкой, и процесс итеративно продолжается до тех пор, пока не станет видимой целевая точка  $T$ . Определим, что точка  $VP_{i+1}$  видна из точки  $VP_i$ , если они могут быть соединены прямым отрезком, который не пересекает ни одного препятствия окружающей среды. После завершения процесса точки множества  $L$  используются как точки для начального сплайна при запуске модифицированного сплайн-метода.

#### 4. Результаты симуляций

Модифицированный алгоритм был реализован в среде Matlab и были проведен ряд виртуальных экспериментов (симуляций). Особое внимание было уделено случаям, когда оригинальный алгоритм [Magid и др., 2006] не мог найти путь для робота. Целевая функция стоимости (уравнение 9) использовалась со следующим набором эмпирически установленных параметров:  $\gamma_1 = 1$ ,  $\gamma_2 = 1$  и  $\gamma_3 = 0,5$ ,  $\gamma_4 = 5$  и  $\gamma_5 = 5$ . Алгоритму удалось найти пути во всех случаях, что явилось естественным следствием того, что начальный сплайн получался из графа Вороного. Граф Вороного считается за приемлемо малое время, которое гораздо меньше времени оптимизации

пути. Например, для среды, показанной на рис. 4 (а) начальный расчет пути на основе графа Вороного занял всего 2 секунды, а для рис. 4 (в) - 6 секунд. Общее время работы нового алгоритма уменьшилось в среднем в три раза по сравнению с исходным алгоритмом. Таким образом, конечный путь рис. 4 (б) был рассчитан всего за 2 итерации в течение 2,5 минут в Matlab, тогда как исходный сплайн-алгоритм провел 18 итераций и 38 минут, чтобы сделать вывод о своей неспособности найти приемлемый путь от точки старта до целевой точки. В другом случае (рис. 1) исходному алгоритму не удалось найти путь, в то время как новый алгоритм на основе Вороного успешно выполнил задачу за 3 итерации и 2 минуты.

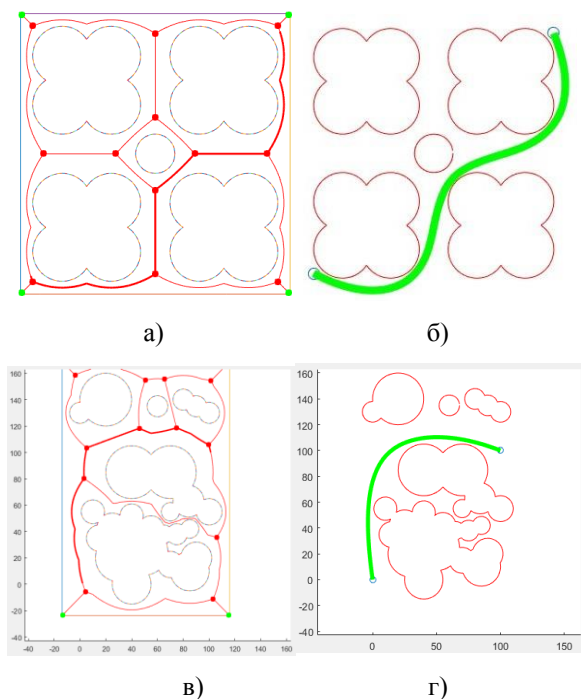


Рис. 4. Результаты планирования пути: решения на графе Вороного (а, в) и результирующие оптимизированные пути (б, г)

## Заключение

В этой статье представлен комбинированный метод расчета плавного и безопасного пути для мобильного робота в статической плоской среде. Предложенная модификация оригинального сплайн-алгоритма помогла

избежать проблемы локальных минимумов и добавила больше гибкости для оптимизации пути. Мы усложнили целевую функцию, введя дополнительные критерии, которые максимизировали время пребывания робота в прямой видимости от начальной и целевой точек. Также в новом алгоритме был использован граф Вороного, благодаря чему мы получили успешную основу для дальнейшей оптимизации пути. В результате проверки алгоритма в Matlab было установлено, что он оказался более успешным и быстреедействующим, чем оригинальный метод. Подробная статистика работы алгоритма в настоящее время в процессе сбора и будет представлена в следующей работе.

В рамках нашей будущей работы мы планируем расширить алгоритм для 3D-среды и добавить новые параметры целевой функции, связанные с 3D-поверхностями. Алгоритм будет объединен в пакет ROS с реализацией на C++ и дополнительно проверен в реальной навигации гетерогенной группы роботов, выполняющей поисково-спасательные задачи.

## Список литературы

- [Magid и др., 2011] E. Magid, T. Tsubouchi, E. Koyanagi, and T. Yoshida. Building a search tree for a pilot system of a rescue search robot in a discretized random step environment. *Journal of Robotics and Mechatronics*, T. 23(4), 2011, с. 567.
- [Pipe и др., 2014] A. Pipe, F. Dailami, and C. Melhuish. Crucial challenges and groundbreaking opportunities for advanced HRI. *IEEE/SICE International Symposium on System Integration (SII)*, 2014, с. 12–15.
- [Rosenfeld и др., 2015] A. Rosenfeld, N. Agmon, O. Maksimov, A. Azaria, and S. Kraus. Intelligent agent supporting human-multi-robot team collaboration. *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, с. 1902-1908.
- [Ronzhin и др., 2016] Ronzhin, I. Vatamaniuk, and N. Pavluk. Automatic control of robotic swarm during convex shape generation. *International Conference and Exposition on Electrical and Power Engineering*, 2016, с. 675–680.
- [Panov и др., 2017] A. I. Panov and K. Yakovlev. Behavior and Path Planning for the Coalition of Cognitive Robots in Smart Relocation Tasks, 2017, с. 3–20.
- [Magid, 2006] E. Magid. Sensor-based robot navigation. Master's thesis, Technion - Israel Institute of Technology, 2006.
- [Buyval и др., 2016] A. Buyval, I. Afanasyev, and E. Magid. Comparative analysis of ROS-based monocular SLAM methods for indoor navigation. *9th International Conference on Machine Vision (ICMV)*, Nice, France, 2016.
- [Khatib и др., 2016] O. Khatib and B. Siciliano, *Springer handbook of robotics*, 2016.
- [Magid и др., 2006] E. Magid, D. Keren, E. Rivlin, and I. Yavneh. Spline-based robot navigation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006. с. 2296-2301.
- [Choset, 2005] H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

- [**Lagarias и др., 1998**] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal on optimization*, T. 9(1), 1998, c. 112-147.
- [**Dijkstra, 1959**] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, vol. 1(1), 1959, c. 269-271.

# ПЛАНИРОВАНИЕ ТРАЕКТОРИИ НА ПЛОСКОСТИ С УЧЕТОМ РАЗМЕРА АГЕНТА (МОБИЛЬНОГО РОБОТА, БЕСПИЛОТНОГО ТРАНСПОРТНОГО СРЕДСТВА)

А.А. Андрейчук (*andreychuk@mail.com*)

РУДН, Москва

К.С. Яковлев (*yakovlev@isa.ru*)

ФИЦ ИУ РАН, ВШЭ, Москва

**Аннотация.** В статье задача планирования траектории на плоскости рассматривается как задача поиска пути на графе особой структуры. Предполагается, что граф является моделью окружающей среды для интеллектуального агента (мобильного робота, беспилотного транспортного средства), который моделируется диском определенного радиуса и может перемещаться на плоскости в произвольном направлении. Предлагаются методы, применение которых позволяет известным алгоритмам эвристического поиска пути на графе учитывать размер агента, проводятся их экспериментальное исследование.<sup>1</sup>

**Ключевые слова:** планирование траектории, эвристический поиск,  $A^*$ ,  $\Theta^*$ , учет размера агента.

## Введение

Планирование траектории – одна из ключевых задач неизбежно возникающих при разработке систем управления беспилотными транспортными средствами, мобильными роботами и т.д. [Макаров и др., 2015]. Известно достаточно много постановок этой задачи и подходов к её решению. Один из наиболее распространенных – представление задачи планирования как задачи поиска пути на графе, вершинам которого соответствуют положения, которые объект управления (в терминологии искусственного интеллекта – интеллектуальный агент) может занимать в пространстве, а ребрам – элементарные траектории между положениями (отрезки прямых или сегменты кривых определенной формы). Обзор графовых моделей, применимых для решения задач планирования приведен в [Яковлев и др., 2013], методов поиска на графах в [Казаков и др.], способов учета различных критериев оптимальности решения в

---

<sup>1</sup> Работа выполнена при финансовой поддержке РФФИ (проект № 17-07-00281).

[Лавренов и др., 2016], результатов применения известных методов для модельных и реальных задач управления беспилотным транспортом в [Афанасьев и др., 2015; Жулев и др., 2013].

В данной работе рассматривается статическая задача планирования траектории на плоскости, а в качестве модели окружающего агента пространства поиска используется граф регулярной декомпозиции, метрический-топологический граф (МТ-граф), известный в англ. терминологии как *grid* [Yap, 2002]. Зачастую при решении задачи планирования траектории с использованием МТ-графа пренебрегают размером и формой агента – см., например, [Андрейчук и др., 2016]. При этом, если граф является только моделью среды и не учитывает свойств агента, то необходима его дополнительная обработка для учета ограничений, накладываемых размером и формой агента.

В робототехнике для учета этих ограничений применяется метод расширения непроходимых областей на величину, соотносящуюся с размером агента [Choset et al., 2005] (см. рис. 1), т.н. преобразование рабочего пространства (*workspace*) в пространство конфигураций (*configuration space* или *C-Space*). Таким образом, при планировании агент может считаться материальной точкой, и найденные пути при этом будут осуществимыми. Однако такой подход не применим для ряда задач, например, для задачи многоагентного планирования, когда имеется множество агентов, и необходимо найти совокупность неконфликтных траекторий для этих агентов. Очевидно, что для определения конфликтов между агентами их нельзя считать лишь материальными точками.

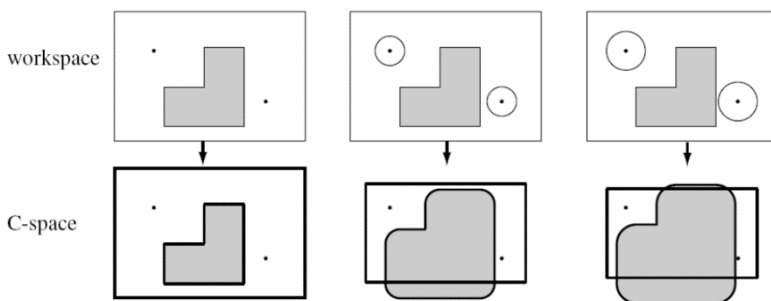


Рис. 1. Учет размера агента за счет расширения непроходимых областей

В данной работе предлагается учитывать ограничения, накладываемые размером агента, в процессе решения задачи планирования. Для этого предлагаются методы, которые необходимо реализовать как составные части известных алгоритмов эвристического поиска, таких как  $A^*$  [Hart et al., 1968],  $\Theta^*$  [Daniel et al., 2010] и др. Предполагается, что агент



представлен открытым диском определенного радиуса, т.к. во многих практических задачах, связанных с автоматизацией управления беспилотных транспортных средств, это предположение уместно (например, когда речь идёт и мультироторных летательных аппаратах или малых колесных роботах).

## 1. Постановка задачи

Агент моделируется открытым кругом радиуса  $r > 0$  и функционирует внутри прямоугольной области на плоскости, которая разбита на квадратные проходимые и непроходимые ячейки, образуя тем самым МТ-граф [Уар, 2002] (см. рис. 2). Размер каждой клетки равен  $l$ . Без ограничения общности будем считать, что  $l = 1$ , а X/Y координаты центров вершин имеют целые значения: 1, 2, 3 и т.д. Начало осей координат находится в нижнем левом углу (см. рис. 2). Положения, в которых может находиться агент, привязаны к центрам проходимых вершин. Более того, агент может находиться в вершине  $U(x, y)$ , только если открытый диск радиуса  $r$  с центром в  $(x, y)$  не задевает какую-либо непроходимую область. Здесь и далее будем называть такие положения (вершины) допустимыми.

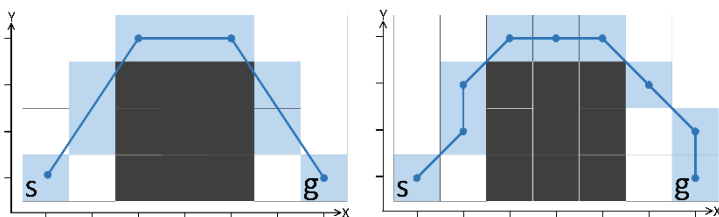


Рис. 2. Пути, построенные алгоритмами Theta\* и A\* для точечного агента. Оба пути являются неосуществимыми для агентов с радиусом  $r > 0$ . Подсвеченные вершины – это вершины, которые идентифицировал алгоритм Брезенхема

Агент может перемещаться вдоль отрезков, которые соединяют центры двух проходимых вершин. При этом движение является допустимым, если открытый диск радиуса  $r$ , двигаясь вдоль секции, не задевает какие-либо непроходимые клетки. Путь между начальным положением (*start*) и целевым (*goal*) – это последовательность смежных допустимых секций. Задача планировщика состоит в отыскании пути из начального положения в целевое. Более короткие пути являются предпочтительными, хотя длина пути не подвержена строгим ограничениям (задача поиска минимального по длине пути не ставится). В качестве базового алгоритма поиска используется Theta\* (модификация известного алгоритма A\*) – см. рис. 2.

## 2. Планирование с учетом размера агента

### 2.1 Theta\*

Theta\* – это эвристический алгоритм поиска, применимый для планирования на МТ-графах и учитывающий возможность перемещения агента в произвольном направлении. Принцип его работы аналогичен алгоритму A\*. Он использует ту же стратегию поиска в пространстве состояний и оперирует списком вершин-кандидатов на дальнейшее рассмотрение, *OPEN*. На каждом шаге алгоритм выбирает наиболее перспективную вершину из списка и раскрывает её. Под раскрытием подразумевается процесс генерации потомков и расчет их *g*-значений (оценок длины пути от начальной вершины). Разница между Theta\* и A\* заключается в том, что при обновлении *g*-значения потомка  $s'$ , при раскрытии вершины  $s$ , Theta\* также проверяет достижимость  $s'$  из родителя  $s$ , т.е.  $parent(s)$ . Если перемещение из  $parent(s)$  в  $s'$  является допустимым, то *g*-значение  $s'$  рассчитывается на основе *g*-значения  $parent(s)$ , иначе оно рассчитывается на основании *g*-значения состояния  $s$ . Более подробно принцип работы алгоритма описан в [Daniel et al., 2010].

Стоит отметить, что мы различаем вершины МТ-графа  $\{C(x, y)\}$  и состояния  $\{s\}$ , хотя между ними есть взаимно-однозначное соответствие. Причина этого различия заключается в том, что в используемой реализации алгоритма Theta\* используются отдельные структуры данных для хранения вершин. Каждое состояние, таким образом, представляет собой кортеж  $s = [cell(s), g(s), parent(s)]$ . В качестве эвристической функции используется Евклидова метрика.

### 2.2 Учет размера агента

#### 2.2.1 Оценка допустимости состояния

Рассмотрим некоторую вершину МТ-графа  $C(x, y)$ , соответствующую преемнику  $s'$ , который находится на рассмотрении на текущем этапе алгоритма. Задача состоит в том, чтобы оценить, может ли агент радиусом  $r$  занимать положение  $(x, y)$ . В случае  $0 < r \leq l/2$  ( $=1/2$ ) достаточно проверить проходимость только вершины  $C$ . Если же радиус агента превышает величину  $l/2$ , требуется более сложная процедура проверки. Сначала строится квадрат, обрамляющий открытый диск радиусом  $r$ , то есть определяются координаты вершин, образующих квадрат. Длина стороны квадрата не превышает  $2 \lfloor r + l/2 \rfloor + 1$ . Затем перебираются ячейки, образующие квадрат, и вычисляется расстояние между центром квадрата  $(x, y)$  и ближайшей к  $(x, y)$  точке, лежащей на границе этой клетки. Такой точкой является либо угол, либо середина стороны, ближайшей к  $(x, y)$  (см. рис. 3). Если расстояние меньше, чем  $r$ , то вершину необходимо проверить

на проходимость. Если все проверенные вершины являются проходимыми, то положение  $(x, y)$  является допустимым.

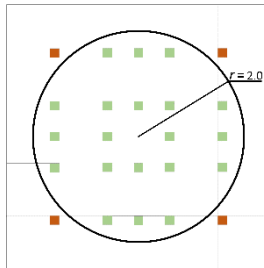


Рис. 3. Оценка допустимости положения агента радиусом  $r = 2,0$ . Вершины, отмеченные зелеными метками, должны быть проверены на проходимость. Положение меток указывает на ближайшие к центру круга точки на границах клеток МТ-графа

### 2.2.2 Оценка допустимости элементарного перемещения

Элементарное перемещение – это переход из проходимой вершины в одну из смежных проходимых вершин. В случае, когда агент переходит в ортогонально смежную вершину, ход является допустимым, если обе вершины являются допустимыми. Если же происходит переход по диагонали, необходимо убедиться, что агент не задевает угол препятствия. Если  $0 < r \leq l/2$ , то необходимо проверить проходимость двух смежных вершин по направлению движения. В случае, когда  $r > l/2$ , агент также может задевать дополнительные клетки и проверка должна осуществляться так, как изложено ниже.

### 2.2.3 Оценка осуществимости неэлементарного перемещения

Неэлементарным перемещением является переход из одной вершины в другую, несмежную с ней. Процедура проверки осуществимости неэлементарного перемещения в литературе обычно называется проверкой линии видимости (в англ. *line-of-sight*) и как правило основывается на использовании алгоритма Брезенхема [Bresenham, 1965], наиболее известного в компьютерной графике алгоритма растеризации прямых линий. На вход алгоритму подаются две пары целых чисел, соответствующие координатам концов отрезка. На выходе получается набор точек с целочисленными координатами (вершин МТ-графа), которые являются ближайшими к этому отрезку. Следует заметить, что использование оригинального алгоритма Брезенхема, неприменимо к рассматриваемой задаче даже в случае, если агент имеет нулевой радиус, т.к. алгоритм не идентифицирует *все* вершины, которые пересекает отрезок, а лишь часть из них (см. рис. 2 слева). В данной работе предлагается

алгоритм проверки линии видимости, применимый для агентов любого неотрицательного радиуса.

Без ограничения общности будем считать, что агент движется слева направо, снизу вверх из клетки  $A(x_A, y_A)$  в клетку  $B(x_B, y_B)$ . Аналогично алгоритму Брезенхема, мы движемся слева направо и обрабатываем каждый столбец МТ-графа. В отличие от алгоритма Брезенхема, который отмечает в каждом столбце только одну вершину (ближайшую к  $\langle A, B \rangle$ ), предлагаемый алгоритм определяет все клетки, которых задевает агент в этом столбце, двигаясь вдоль секции  $\langle A, B \rangle$ . Далее подробно опишем процесс определения задеваемых клеток, которые находятся над  $\langle A, B \rangle$ . Определение задеваемых клеток, которые находятся под секцией, осуществляется аналогично.

Рассмотрим ситуацию, когда агент проходит через столбец  $i$ , а ближайшей вершиной, определенной алгоритмом Брезенхема, является  $U(i, j)$ . На каждом шаге алгоритм вычисляет значение ошибки  $error(i)$ , которая фактически является разницей между  $j$  и реальным значением секции  $\langle A, B \rangle$  вдоль оси  $Y$ , в точке с той же самой  $X$ -координатой, то есть  $i$ . Далее рассмотрим величину  $offset(i) = error(i) + \Delta_y/2\Delta_x + |AB'|$  (см. рис. 4), где  $\Delta_x = |x_B - x_A|$ ,  $\Delta_y = |y_B - y_A|$ . Очевидно, что точка с координатами  $(i+1/2, j+offset)$  – это самая верхняя точка в столбце  $i$ , которую задевает агент. Величина  $AB'$  эквивалентна значению  $r \cdot \sqrt{(\Delta_x^2 + \Delta_y^2)}/\Delta_x$ , т.к. треугольники  $ABC$  и  $AB'C'$  являются подобными. Зная  $offset(i)$ , можно вычислить число клеток над  $U(i, j)$ , которые необходимо проверить на проходимость:  $N_{u-hi}(i) = \lfloor offset(i) + 1/2 - \varepsilon \rfloor$ , где  $\lfloor \cdot \rfloor$  - округление до целого вниз, а добавление  $\varepsilon \ll 1/2$  необходимо для корректного округления в случаях, когда тело агента находится строго на границе между клетками.

Начало отрезка  $A$  является особым случаем. В общем случае, агент может задевать дополнительные вершины даже в тех столбцах, которые предшествуют ряду  $x_A$ . Число таких столбцов,  $k$ , зависит от величины  $C'D$ , которая равна  $r \cdot \Delta_y / \sqrt{(\Delta_x^2 + \Delta_y^2)}$ , т.к. треугольники  $ABC$  и  $ADC'$  являются подобными. Таким образом,  $k = \lfloor C'D \rfloor + 1/2 - \varepsilon - 1$ , где “-1” необходимо для того, чтобы не учитывать столбец  $x_A$ .

Конечное положение  $B$  также является особым случаем. Как можно заметить на рис. 4 число вершин, которые задевает агент, когда приближается к конечному положению, меньше чем величина  $N_{u-hi}(i)$ . В то же время, очевидно, что вершины в этих столбцах уже были проверены, когда проверялась осуществимость положения  $(x_B, y_B)$ . Поэтому, их проверять не нужно.

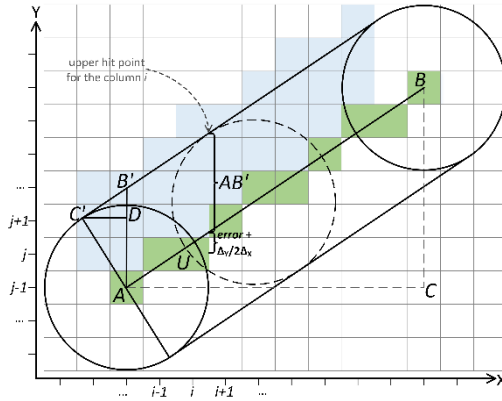


Рис. 4. Определение клеток (вершин МТ-графа), задеваемых агентом во время совершения неэлементарного перемещения. Зеленым отмечены вершины, которые нашел алгоритм Брезенхема. Голубым отмечены вершины, найденные предлагаемым методом

Псевдокод предложенного алгоритма проверки линии видимости представлен на рис. 5. Часть кода, отвечающая за случай, когда  $\Delta_y \geq \Delta_x$ , опущена, т.к. все операции и проверки являются аналогичными. Для оптимизации кода все операции по возможности были сведены к целочисленным вычислениям.

Algorithm 1: Line Of Sight	
Input: $s_x, s_y, g_x, g_y, r$	18 for $x$ from $s_x$ to $g_x$ do
1 $\Delta_x :=  s_x - g_x $ , $\Delta_y :=  s_y - g_y $ , error := 0	19 if $cell\_is\_obstacle(x, y)$ then
2 $gap := \lceil r \cdot \sqrt{\Delta_x^2 + \Delta_y^2} + (\Delta_x + \Delta_y)/2 - \epsilon \rceil$	20   return false
3 if $\Delta_x > \Delta_y$ then	21 if $x < g_x - extra\_check$ then
4   if $s_x > g_x$ then	22   for $k$ from 1 to $\lfloor (gap - error)/\Delta_x \rfloor$ do
5     $swap(s_x, g_x)$ , $swap(s_y, g_y)$	23     if $cell\_is\_obstacle(x, y + k \cdot step_y)$ then
6   $step_y := sign(g_y - s_y)$	24       return false
7   if $step_y = 0$ then	25 if $x > s_x + extra\_check$ then
8     $step_y := 1$	26   for $k$ from 1 to $\lfloor (gap + error)/\Delta_x \rfloor$ do
9   $extra\_check := \lfloor \frac{r \cdot \Delta_y}{\sqrt{(\Delta_x^2 + \Delta_y^2)}} + 1/2 - \epsilon \rfloor$	27     if $cell\_is\_obstacle(x, y - k \cdot step_y)$ then
10 for $k$ from 1 to $extra\_check$ do	28       return false
11   error := error + $\Delta_y$	29 error := error + $\Delta_y$
12   for $n$ from 1 to $\lfloor (gap - error)/\Delta_x \rfloor$ do	30 if $2 \cdot error > \Delta_x$ then
13     if $cell\_is\_obstacle(s_x - k, s_y + n \cdot step_y)$ then	31     $y := y + step_y$
14       return false	32     error := error - $\Delta_x$
15     if $cell\_is\_obstacle(g_x + k, g_y - n \cdot step_y)$ then	33 else
16       return false	34   the same, but all checks and operations
17 error := 0, $y := s_y$	35   should be swapped between X- and Y-axis
	36 return true

Рис. 5. Псевдокод предлагаемого алгоритма проверки линии видимости

### 3. Экспериментальные исследования

Для проведения экспериментальных исследований был реализован алгоритм Theta\* (на языке C++), содержащий описанные выше проверки. Эксперименты проводились на персональном компьютере (AMD FX-8350 (4.0 GHz), 16 Gb RAM) под управлением ОС Windows-8.1.

В качестве входных данных использовались 2 карты из открытой коллекции MovingAI [Sturtevant, 2012] размером 512 x 512 клеток. Первая карта, *AR0700SR*, представляет собой открытую местность, разделенную на несколько областей, соединенных проходами различной ширины. Вторая карта, *Rooms*, состоит из комнат размером 32 x 32, которые изначально изолированы. Затем добавлены 255 дверей шириной в 5 клеток таким образом, чтобы они соединяли все 256 комнат между собой. После этого дополнительно добавлены 32 двери шириной в 5, 32 двери шириной в 3 и 32 двери шириной в 1 клетку соответственно.

Радиус агента составлял:  $r = 0,5; 0,7; 1,0; 2,0$ . Также для сравнения был протестирован точечный агент без тела ( $r = 0$ ).

Задания генерировались следующим образом. Начальное положение выбиралось случайным образом (недопустимые положения отбрасывались). После этого выбиралось целевое положение, таким образом, чтобы, во-первых, оно было допустимым, во-вторых, путь, находимый A\*, имел длину в диапазоне [220; 240] (это необходимо для обоснованного усреднения результатов). Всего было сгенерировано 1500 заданий для карты из *Baldur's Gate* и 3000 заданий для карты *Rooms*.

Результаты исследований представлены в таблицах 1 и 2. При усреднении учитывались только те задания, решения для которых были найдены для агентов всех размеров. Значение числа итераций алгоритма можно также интерпретировать как показатель расхода памяти, т.к. на каждом шаге алгоритм раскрывает одну вершину и сохраняет её в списке до окончания работы. Первый ряд в таблицах,  $r = 0(Br)$ , относится к алгоритму Theta\* с проверкой линии видимости, использующей оригинальный алгоритм Брезенхема, который может находить пути неосуществимые даже для точечного агента.

Таблица 1

Результаты тестирования на карте *AR0700SR*

	<i>Time (ms)</i>	<i>Path length</i>	<i>Steps</i>	<i>SR</i>
$r = 0(Br)$	1.324	228.95	5 660	100.0%
$r = 0$	1.817	229.18	5 688	100.0%
$r = 0.5$	2.009	229.71	5 787	100.0%
$r = 0.7$	2.109	231.61	5 483	96.1%

$r = 1.0$	2.249	232.86	5 433	94.8%
$r = 2.0$	3.794	262.45	7 206	84.3%

Результаты тестирования на карте *AR0700SR* показывают, что для агентов, чей размер превышает одну клетку МТ-графа ( $r > 0,5$ ), решение некоторых заданий может не существовать. Причина этого заключается в том, что начальные и целевые положения в этих заданиях расположены в разных областях, соединенных между собой лишь узкими проходами, через которые не могут пройти агенты большого размера. По той же причине с увеличением размера агента увеличивается и средняя длина пути. Также можно заметить, что число итераций у агентов с радиусом 0,7 и 1,0 меньше, чем итераций у агентов меньших размеров. Объяснить это можно тем, что проходимые вершины смежные с препятствиями являются недопустимыми для агентов большого размера, благодаря чему пространство поиска для них сокращается. С увеличением размера агента время работы закономерно растет, т.к. чем больше агент, тем больше времени требуется для проверки линии видимости.

Таблица 2

Результаты тестирования на карте *Rooms*

	<i>Time (ms)</i>	<i>Path length</i>	<i>Steps</i>	<i>SR</i>
$r = 0(\text{Br})$	1.408	218.22	6 339	100%
$r = 0$	1.829	218.73	6 421	100%
$r = 0.5$	1.939	219.29	6 534	100%
$r = 0.7$	5.296	285.26	15 616	100%
$r = 1.0$	5.705	287.25	15 745	100%
$r = 2.0$	6.519	320.84	16 073	100%

В таблице 2 представлены результаты экспериментов на карте *Rooms*. Аналогично результатам, полученным на карте *AR0700SR*, увеличение размера агента приводит к увеличению времени работы алгоритма, средней длины пути и числу раскрытых вершин. При этом разница на этой карте более существенная. Например, время планирования для агента с  $r = 2,0$  в среднем в 3,4 раза больше, чем для агента с радиусом 0,5. Также у агента самого большого размера длина пути в среднем в 1,5 раза выше, а число требуемых итераций увеличивается на 46%.

## Выводы

Результаты экспериментов подтверждают гипотезу о том, что учет размера агента существенно изменяет поведение алгоритма планирования

(Theta\* в рассматриваемом случае). Этот факт определенно должен быть принят во внимание при применении хорошо зарекомендовавших себя эвристических методов для решения задач планирования для реальных физических объектов, имеющих определенную форму и размер.

## Список литературы

- [Андрейчук и др., 2016] Андрейчук А.А., Яковлев К.С. Метод разрешения конфликтов при планировании пространственных траекторий для группы беспилотных летательных аппаратов // Третий Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта». 2016. С. 31-40.
- [Афанасьев и др., 2015] Афанасьев И.М., Сагитов А.Г., Данилов И.Ю., Магид Е.А. Навигация гетерогенной группы роботов (БПЛА и БНР) через лабиринт в 3D симуляторе Gazebo методом вероятностной дорожной карты // Второй Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта». 2015. С. 18-25.
- [Жулев и др., 2013] Жулев В.И., Левушкин В.С., Нгуен Т.Н. Планирование локальной траектории автомобиля-робота в реальном времени // Вестник РГРТУ, 4(46), 2013. С. 18–23.
- [Казаков и др., 2016] Казаков К.А., Семенов В.А. Обзор современных методов планирования движения // Труды ИСП РАН, 28(4), 2016. С. 241-294
- [Лавренов и др., 2016] Лавренов Р.О., Афанасьев И.М., Магид Е.А. Планирование маршрута для беспилотного наземного робота с учетом множества критериев оптимизации // Третий Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта». 2016. С. 10-20.
- [Макаров и др., 2015] Макаров Д.А., Панов А.И., Яковлев К.С. Архитектура многоуровневой интеллектуальной системы управления беспилотными летательными аппаратами // Искусственный интеллект и принятие решений, 3, 2015. С.18-32.
- [Яковлев и др., 2013] Яковлев К.С., Баскин Е.С. Графовые модели в задаче планирования траектории на плоскости // Искусственный интеллект и принятие решений, 1, 2013. С.5-12.
- [Bresenham. 1965] Bresenham, J. E. Algorithm for computer control of a digital plotter. *IBM Systems journal*. 1965. № 4(1), P.25-30.
- [Choset et al., 2005] Choset, H., Burgard, W., Hutchinson, S., Kantor, G., Kavradi, L. E., Lynch, K., and Thrun, S. Principles of Robot Motion: Theory, Algorithms, and Implementation, MIT Press, April 2005.
- [Daniel et al., 2010] Daniel, K., Nash, A., Koenig, S., and Felner, A. Theta\*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research*. 2010. № 39, P.533-579.
- [Hart et al., 1968] Hart, P. E., Nilsson, N. J., and Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*. 1968. № 4(2), P.100-107.



- [**Sturtevant, 2012**] Sturtevant, N. Benchmarks for grid-based pathfinding. *IEEE Transactions on Computational Intelligence and AI in Games*. 2012. №4(2), P.144–148.
- [**Yap, 2002**] Yap, P. Grid-based path-finding // In Proceedings of 15th Conference of the Canadian Society for Computational Studies of Intelligence, Springer: Berlin, Heidelberg. 2002. P.44-5

УДК 681.51

## ЗАДАЧА ОБУЧЕНИЯ ДВИЖЕНИЮ ПО ТРАЕКТОРИИ БЕСПИЛОТНОГО ТРАНСПОРТНОГО СРЕДСТВА С ИСПОЛЬЗОВАНИЕМ ДСМ-МЕТОДА

Д.А. Добрынин (*minirobot@yandex.ru*)  
ФИЦ ИУ РАН, Москва

**Аннотация.** В статье описывается задача планирования движения мобильного робота по траектории при наличии препятствий с использованием динамического ДСМ-метода. В качестве учителя может выступать человек или любой внешний алгоритм. Приведены качественные результаты процесса обучения.

**Ключевые слова:** ДСМ-метод, машинное обучение, движение по траектории.

### Введение

Одной из важных задач планирования движения мобильных роботов является задача достижения определенной цели при наличии препятствий. Несмотря на кажущуюся внешнюю простоту, решение данной задачи вызывает большие сложности при попытках реализации для беспилотных транспортных средств (БТС). Эта задача может быть решена различными методами: от построения траектории математическими методами до различных методов ИИ. Так как робот является системой реального времени, то естественно, что при реализации предпочтение будет отдаваться методам, обладающим наименьшей вычислительной сложностью, при сохранении приемлемого качества управления. Одним из перспективных являются методы, основанные на обучении, позволяющие получить требуемый результат и обойтись без построения сложных математических моделей [Anshakov, 2010].

Современные БТС являются сложными системами, имеющими большое количество датчиков (рецепторов) и управляемых механизмов (эффекторов). Увеличение количества сигналов рецепторов приводит к очевидным сложностям обучения. Например, при  $R$  независимых входных сигналах размерность входного алфавита автоматной модели определяется как  $\dim X=2^R$  и очень быстро растет при увеличении  $R$ . Введение параллелизма, например, в случае использования нейронной сети, не спасает ситуацию - время обучения возрастает экспоненциально и

становится недопустимо большим даже для сравнительно небольших выборок.

Отсюда неизбежно возникает задача классификации множества входных сигналов, т.е. распознавание ситуаций. Классификатор может быть различного вида – от множества продукций до реализации в виде нейронной сети или хромосомы генетического алгоритма. Его функция заключается в анализе входного вектора и определении класса, к которому этот вектор относится.

Основная задача состоит в создании этого классификатора как результата адаптивного поведения робота, т.е. результата его обучения или самообучения.

На работе «Амур» [Добрынин и др., 2006] проводились эксперименты по построению классификатора с помощью двух методов: эволюционного моделирования (ЭМ) и динамического ДСМ-метода [Добрынин, 2006]. В качестве простого примера было взято движение робота по контрастной полосе, когда робот имеет фотодатчики и двигается по полосе. Требовалось построить управляющую систему, которая бы заставляла робота корректно двигаться по этой полосе.

## 1. Постановка задачи

В данной работе рассматривается следующий вариант задачи движения по траектории, представленный на рис. 1:

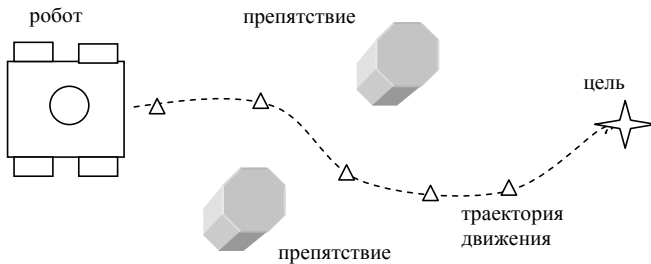


Рис. 1. Движение робота по траектории с препятствиями

Беспилотное транспортное средство стремится достичь цели, минуя препятствия. Точки траектории при движения робота могут задаваться как специальным приемником, так и заданием координат относительно робота. Для этого на роботе должен быть либо приемник, который умеет

определять направление на текущую цель с некоторой точностью, либо система определения координат. При движении к цели робот должен объехать препятствие (или несколько препятствий), которые могут динамически менять свое местоположение. Для обнаружения препятствия на роботе установлена система обнаружения, которая определяет направление на препятствие и дальность до него (до ближайшей точки препятствия) с некоторой точностью.

Особенностями данной модели являются:

- траектория движения задается явным образом;
- существуют неустранимые погрешности измерения углов, координат и расстояний, определяемые техническими параметрами устройств обнаружения;
- препятствия могут перемещаться, поэтому требование оптимальности траектории движения робота не ставится.

## 2. Динамический ДСМ-метод

Для создания системы управления, способной обучаться, можно построить классификатор входных сигналов с помощью ДСМ метода. Аббревиатура ДСМ расшифровывается как Джон Стюарт Милль. Метод назван в честь британского философа XIX века, идеи которого положены в основу метода. ДСМ-метод автоматического порождения гипотез [Финн, 1991] является теорией автоматизированных рассуждений и способом представления знаний для решения задач прогнозирования в условиях неполноты информации. Классический ДСМ метод работает с замкнутым множеством исходных примеров, которое формируется экспертом и составляет базу знаний. Каждый пример описывается множеством элементарных признаков и наличием (или отсутствием) целевого свойства. С помощью специальных логических процедур из этой базы знаний ДСМ-система получает гипотезы, которые объясняют свойства исходных примеров из-за наличия или, наоборот, отсутствия в структуре примеров определенной совокупности признаков. Таким образом, ДСМ система выделяет из исходной информации в базе знаний существенные совокупности признаков, т. е. осуществляет автоматическую классификацию. ДСМ метод успешно применим в тех областях знаний, где пример можно представить в виде множества (или кортежа) элементарных признаков.

В отличие от классического ДСМ метода, который работает с замкнутым множеством исходных примеров и заранее определенными их свойствами, динамический ДСМ метод позволяет работать в открытой среде с неизвестным заранее количеством примеров [Добрынин, 2006].

Динамический ДСМ работает в двух режимах:

- режим обучения, когда происходит заполнение базы фактов (множество обучающих примеров) и генерируются гипотезы, составляющие базу знаний;

- рабочий режим, когда полученные ранее гипотезы используются для выработки сигналов управления.

Множество обучающих примеров – это множество пар вида

$$E = \{e_i\} = \{(X_i, u^i)\},$$

где  $X_i$  - вектор сигналов рецепторов,  $u^i$  – вектор управления (состояние исполнительных механизмов). Элементы векторов сигналов и управления представляются парами двоичных значений:

включено = {01},      выключено = {10}

не важно = {00}

Такое представление необходимо для корректного выполнения операций пересечения и вложения над битовыми строками.

Операция пересечения (нахождения общей части) двух объектов при использовании битовых строк реализуется с помощью логической функции «побитовое И». Операция вложения, отвечающая на вопрос – входят ли все компоненты объекта А в объект В, реализуется как «побитовое И» элементов объектов А и В, а затем сравнение результата с элементами вкладываемого объекта А.

Гипотезы представляются в виде множества пар вида:

$$G = \{g_i\} = \{(x_i, y^i)\},$$

где  $x_i$  – часть вектора сигналов рецепторов,  $y^i$  – требуемый вектор управления (необходимое действие). Гипотезы существуют двух видов: положительные гипотезы определяют, при каком входном воздействии выполнять действия, приводящие к положительному результату; отрицательные гипотезы определяют, какие действия не нужно делать при данном входном воздействии.

В режиме обучения для формирования обучающих примеров используется внешний алгоритм – так называемый «учитель». Данный алгоритм получает на вход информацию от рецепторов и вырабатывает управляющие сигналы, необходимые для адекватного поведения робота. Совокупность сигналов рецепторов и выработанных для них управляющих воздействий определяет один обучающий пример. Этот пример проверяется на уникальность и заносится ДСМ системой в базу фактов. После занесения каждого нового примера во множество обучающих примеров производится поиск гипотез. На полученные гипотезы могут накладываться дополнительные ограничения, например, запрет на контрпримеры, когда положительная гипотеза не должна вкладываться в отрицательные примеры и наоборот. Эти ограничения определяются используемым ДСМ-методом [Кузнецов, 1991].

Полученное множество гипотез будет содержать все возможные пересечения (общие части) обучающих примеров. Далее среди них отбираются минимальные гипотезы, то есть такие, которые вкладываются в остальные. Тем самым количество «полезных» гипотез резко сокращается. Полученные минимальные гипотезы проверяются на уникальность и заносятся в базу знаний.

Обучение должно производиться до тех пор, пока база знаний не перестанет пополняться новыми гипотезами. Очевидно, что в этом случае обучающий алгоритм перебрал все возможные варианты входных воздействий, на которые он способен реагировать, и можно считать, что база фактов достаточно полна.

В рабочем режиме ДСМ система получает на вход сигналы рецепторов, из которых формируется тестовый вектор. Принятие решения происходит путем проверки вложения гипотез в этот вектор. Если в тестовый вектор сигналов рецепторов вкладывается гипотеза, то робот должен действовать в соответствии с ней. Если же ни одной гипотезы не найдено, то это неизвестное состояние, для которого нужно сформировать случайный вектор управления (или ничего не делать, например).

### 3. Обучение движению

Для обучения движению ДСМ-системы необходима внешняя система управления (учитель), которая принимает информацию от датчиков положения, например, о местоположении текущей точки траектории движения, информацию от датчика препятствий и формирует управляющие воздействия (управляет двигателями) как показано на рис. 2.

В качестве такого учителя может выступать как сам человек, так и любая другая управляющая система. Например, в работе [Добрынин, 2009] рассматривалось обучение с помощью нечеткой системы управления.

Основным требованием к обучающей системе управления является непротиворечивость команд управления. В противном случае, такие противоречивые команды могут «ввести в заблуждение» обучаемую систему, что приводит к игнорированию противоречивых входных сигналов, и соответственно, снижению качества обучения.

Сам процесс обучения ДСМ-системы выглядит следующим образом:

- «учитель» получает входную информацию и вырабатывает управляющие воздействия. Вся эта информация подается в ДСМ-систему, при этом формируется т.н. «обучающий пример», который ДСМ-система заносит в базу фактов. Если в этой базе фактов такой пример уже есть, то ничего не происходит.

- если появляется новый обучающий пример, ранее не встречающийся в базе фактов, то в этом случае он передается ДСМ-решателю, который

формирует с его помощью новую гипотезу. Если полученная гипотеза удовлетворяет критериям непротиворечивости, то она добавляется в базу гипотез.

- пополнение базы фактов и получение новых гипотез производится до тех пор, пока работает режим обучения.

Критерием завершения обучения может служить тот факт, что перестает пополняться база гипотез. Это означает, что на вход обучаемой системы не поступает новая информация. После окончания режима обучения ДСМ-система имеет набор гипотез, которые в дальнейшем используются для работы обученной системы управления.

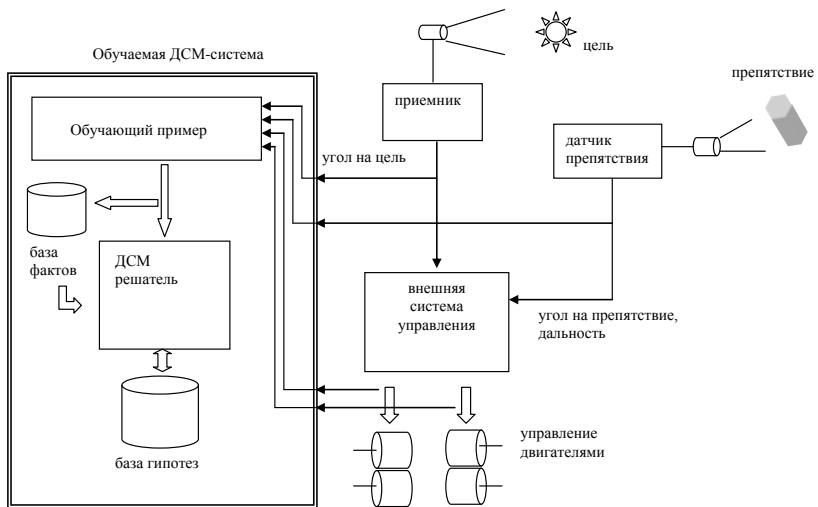


Рис. 2. Обучение ДСМ-системы

Работа обученной таким способом системы управления описана выше.

Эксперименты на компьютерных симуляторах и натурные эксперименты на мобильных роботах показывают, что внешние отличия в характере движения робота между работой системы-учителя и обученной с ее помощью ДСМ-системы практически отсутствуют. Это говорит о высоком качестве работы обученной системы управления.

## Заключение

По результатам экспериментов выявились следующие отличия метода эволюционного моделирования (ЭМ) и ДСМ-метода:

**Достаточность обучающих примеров.** При наличии представительной выборки обучающих примеров оба метода дают хорошие результаты. Однако в условиях неполноты обучающего множества метод ЭМ дает более устойчивые результаты по сравнению с ДСМ. Это связано прежде всего с характером обучения.

**Непротиворечивость обучающей выборки.** ДСМ, в отличие от ЭМ, плохо применим в условиях противоречий в обучающих примерах. Такая ситуация может возникать, когда учитель ошибается в оценке состояния датчиков. Ошибки такого типа необходимо отсеивать на этапе формирования обучающих примеров. В ЭМ подобная противоречивость не так критична, т.к. она приводит в худшем случае к неопределенности фенотипического поведения.

**Эффективность обучения (скорость).** Обучение в ЭМ – принципиально длительный процесс. Для устойчивого обучения методом эволюционного моделирования иногда требуются сотни тысяч тактов. В этом отношении ДСМ-метод обладает несомненным преимуществом - для обучения достаточно получить несколько разных обучающих примеров. В экспериментах роботу достаточно было проехать один круг на реальном полигоне чтобы сформировались все необходимые гипотезы.

**Динамическое обучение.** Теоретически ЭМ может работать и в открытой среде с неизвестным заранее количеством примеров, практически же это связано с большими вычислительными затратами. Динамический ДСМ метод позволяет эффективно работать с заранее неизвестным количеством примеров при сравнительно небольших вычислительных затратах.

**Обучение с учителем.** Все рассматриваемые алгоритмы обучения работают с «учителем», который фактически формирует для них представительную обучающую выборку. В качестве «учителя» в проводимых экспериментах выступал внешний алгоритм управления роботом. В принципе, учителем может быть и человек, при условии непротиворечивости выдаваемых им управляющих воздействий.

**Требуемые ресурсы.** При реализации практических алгоритмов встает проблема ограниченности вычислительных ресурсов автономного робота. Если моделирование эволюции требует весьма больших временных и емкостных затрат, то для работы ДСМ метода достаточно небольших вычислительных ресурсов.

Эти особенности, особенно высокая скорость обучения и нетребовательность к вычислительным ресурсам, позволяют выделить динамический ДСМ-метод как один из основных претендентов для построения обучаемой системы управления для роботов. Ограничением использования данного метода является необходимость представления входных данных в виде дискретного множества. Разбиение на интервалы



исходных непрерывных значений позволяет только частично решить эту проблему, поскольку для увеличения точности необходимо уменьшать шаг дискретизации, а при этом быстро растет размерность внутреннего представления данных и снижается быстродействие.

## Список литературы

- [Anshakov, 2010] Anshakov O, Gergely T. Cognitive Reasoning: A Formal Approach. Springer, 2010
- [Добрынин, 2006] Добрынин Д.А. Динамический ДСМ-метод в задаче управления интеллектуальным роботом.// Десятая национальная конференция по искусственному интеллекту КИИ-2006, 25-28 сентября 2006 г., Обнинск, Труды конференции, М:Физматлит 2006, т.2.
- [Добрынин и др., 2006] Добрынин Д.А., Карпов В.Э. Моделирование некоторых форм адаптивного поведения интеллектуальных роботов.// Информационные технологии и вычислительные системы, Москва, №2, 2006.
- [Добрынин, 2009] Волкова Т.А., Добрынин Д.А. Сравнение системы нечеткого вывода и обучаемой ДСМ-системы при планировании движения мобильного робота// V-международная научно-практическая конференция «Интегрированные модели и мягкие вычисления», 20-30 мая 2009 г., г. Коломна, М.:Физматлит, 2009., Т1.
- [Кузнецов, 1991] Кузнецов С.О. ДСМ-метод как система автоматизированного обучения // Итоги науки и техники. Сер. «Информатика». Т. 15. - М: ВИНТИ, 1991.
- [Финн, 1991] Финн В.К. Правдоподобные рассуждения в интеллектуальных системах типа ДСМ //Итоги науки и техники. Сер. «Информатика». Т. 15. - М.: ВИНТИ, 1991.

УДК 519.878, 004.02

## ПЛАНИРОВАНИЕ МАРШРУТА ПОИСКА ДЛЯ АВТОНОМНОГО БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА С ИСПОЛЬЗОВАНИЕМ ЭНТРОПИЙНОГО ПОДХОДА

Н. А. Михайлов (*mikhailov.mai@gmail.com*)  
Московский авиационный институт (национальный  
исследовательский университет) (МАИ)

**Аннотация.** В работе представлен алгоритм планирования траектории движения автономного беспилотного летательного аппарата для решения задачи поиска объектов. Алгоритм основан на вычислении энтропийной карты поиска. Критерием выбора пути выступает максимальная информационная пропускная способность. На основе предложенных информационных моделей, было проведено компьютерное моделирование поиска. Результаты моделирования подтверждают работоспособность предложенного подхода.

**Ключевые слова:** беспилотный летательный аппарат, групповое применение, поиск, энтропия, информативность признака.

### Введение

Задача оптимального планирования траектории движения для одиночного или группы автономных роботов (в частности, для беспилотных летательных аппаратов (БЛА)) является актуальной на сегодняшний день [Лавренов и др., 2016], [Андрейчук и др., 2016]. Сложность подобных задач связана, в первую очередь, с многокритериальной оптимизацией, учитывающей разные по своей природе факторы. Как правило при решении точка пространства, в которую следует доставить автономный робот определена. В данной работе рассматривается одно из направлений подобных задач – это планирование траектории поиска наземных объектов. Специфика задачи планирования траектории поиска состоит в том, что положение объекта поиска заранее не известно. Классические алгоритмы планирования траектории поиска учитывают лишь вероятность нахождения объекта поиска в заданной области [Абчук и др. 1977]. В работе предлагается подход, учитывающий не только вероятность нахождения объекта поиска в заданной области, но

и его наблюдаемость, определяемую аппаратурой и условиями наблюдения [Ким и др. 2002].

## Постановка задачи

Пусть имеется карта местности, на которой находится объект интереса (ОИ) –  $o$  (объект – object). Для простоты представим карту в виде сетки размером  $M \times N$ , где  $M$  – количество строк разбиения, а  $N$  – количество столбцов. Для каждой ячейки сетки имеется, некоторая априорная информация о вероятности присутствия ОИ в ней  $P(o)$ , причем суммарная вероятность нахождения объекта в заданной области равна:

$$\sum_{m=1}^M \sum_{n=1}^N P(o_{m,n}) = 1, \quad (1)$$

где  $m, n$  – координаты ОИ.

Вне зависимости от физической природы носителя сообщений и структуры признаков, оценка их информативности рассчитывается одинаково [Ким 2001]. Поэтому, для обнаружения ОИ задан набор признаков  $\bar{u} = \{u_1 \in U_1, \dots, u_R \in U_R\}$ , где  $R$  общее количество всех признаков ОИ (яркость, длина, цвет, форма и т.д.), а также определены условные плотности распределения вероятности возникновения признака  $u_r$  в каждой ячейке карты, при условии нахождения в этой ячейке ОИ или его отсутствия:  $p(u_r | o_{m,n})$  и  $p(u_r | b_{m,n})$ , где за  $b$  обозначено событие отсутствия ОИ в ячейке (фон – background). Заданные плотности распределения определяют наблюдаемость объекта в некоторой области.

Требуется построить траекторию движения каждого БЛА, при которой достигается его максимальная информационная пропускная способность поиска.

## 1. Энтропийное описание задачи

В данной работе за количественную меру информации  $I$  принята шенноновская мера. При не равновероятных исходах  $P_k$ , где  $P_k$  – вероятность  $k$ -ого исхода из  $K$  возможных. При этом неопределенность (энтропия)  $H$ , определяется той же самой мерой, что и информация [Коган 1981]:

$$H = -\sum_{k=1}^K P_k \log_2 P_k \quad (2)$$

Количественная мера информации, приносимая сообщением  $y$  о событии  $x$ , в наиболее общем случае, определяется выражением:

$$I(y|x) = H(x) - H(x|y), \quad (3)$$

где  $H(x)$  – неопределенность ситуации  $x$  до события  $y$ ;

$H(x|y)$  – неопределенность ситуации  $x$ , оставшаяся после наступления события  $y$ .

Информационная пропускная способность  $C$  обозначает скорость переработки информации  $I$  за время  $T$ :

$$C = \frac{I}{T} = \frac{H(x) - H(x|y)}{T}. \quad (4)$$

Так как  $T$  включает время полета БЛА к выбранной ячейке, то, следовательно, пропускная способность учитывает расстояние до этой ячейки.

При решении задачи поиска в заданной области количество возможных исходов  $K = M \cdot N$ , т.е. определяется количеством ячеек сетки.

При решении задачи поиска БЛА имеет на борту комплексную систему наблюдения и решает с ее помощью задачу обнаружения. Задача обнаружения является частным случаем задачи распознавания, она характеризуется наличием двух различных классов объектов: ОИ и фон ( $\Omega = 2$ ).

Если дискретизировать область значений некоторого  $r$ -ого признака на  $Q$  отсчетов, то общее количество полезной информации при распознавании  $\Omega$  классов, которое может быть получено при измерении всех значений  $r$ -ого признака в некоторой ячейке с координатами  $m, n$  равно [Ким 2001]:

$$I_{m,n} = H_0(\Omega) - H(\Omega|U_r) = -\sum_{k=1}^K \sum_{\omega=1}^{\Omega} P_k(\omega) \log_2 P_k(\omega) + \\ + \sum_{\alpha=1}^{\Omega} P(\alpha_{m,n}) \sum_{\beta=1}^{\Omega} \sum_{q=1}^Q P(u_{r,q} | \alpha_{m,n}) \cdot P(\beta_{m,n} | u_{r,q}) \log_2 P(\beta_{m,n} | u_{r,q}) \quad (5)$$

где  $P_k(\omega)$  – априорная вероятность  $k$ -ого исхода  $\omega$  класса;

$P(\alpha_{m,n})$  – априорная вероятность наличия (присутствия) класса  $\alpha$  в ячейке с координатами  $m, n$ ;

$P(u_{r,q} | \alpha_{m,n})$  – вероятность приема  $q$ -ого значения  $r$ -ого признака при условии наличия класса  $\alpha$  в ячейке с координатами  $m, n$ ;

$P(\beta_{m,n} | u_{r,q})$  – апостериорная вероятность наличия класса  $\beta$  в ячейке с координатами  $m, n$  при условии приема  $q$ -ого значения  $r$ -ого признака, может быть рассчитана по формуле Байеса:

$$P(\beta_{m,n} | u_{r,q}) = \frac{P(\beta_{m,n})P_j(u_{r,q} | \beta_{m,n})}{\sum_{\omega=1}^{\Omega} P(\omega_{m,n})P_j(u_{r,q} | \omega_{m,n})}, \quad (6)$$

Соотношение (5) позволяет связать априорную информацию о вероятности нахождения ОИ в заданной области, и его наблюдаемость в каждой конкретной ячейке. Причем, если в ячейке условные плотности распределения признака для всех классов  $\Omega$  не пересекаются, то  $I_{m,n}$  полностью определяется начальной энтропией  $H_0(\Omega)$ .

## 2. Алгоритм построения маршрута

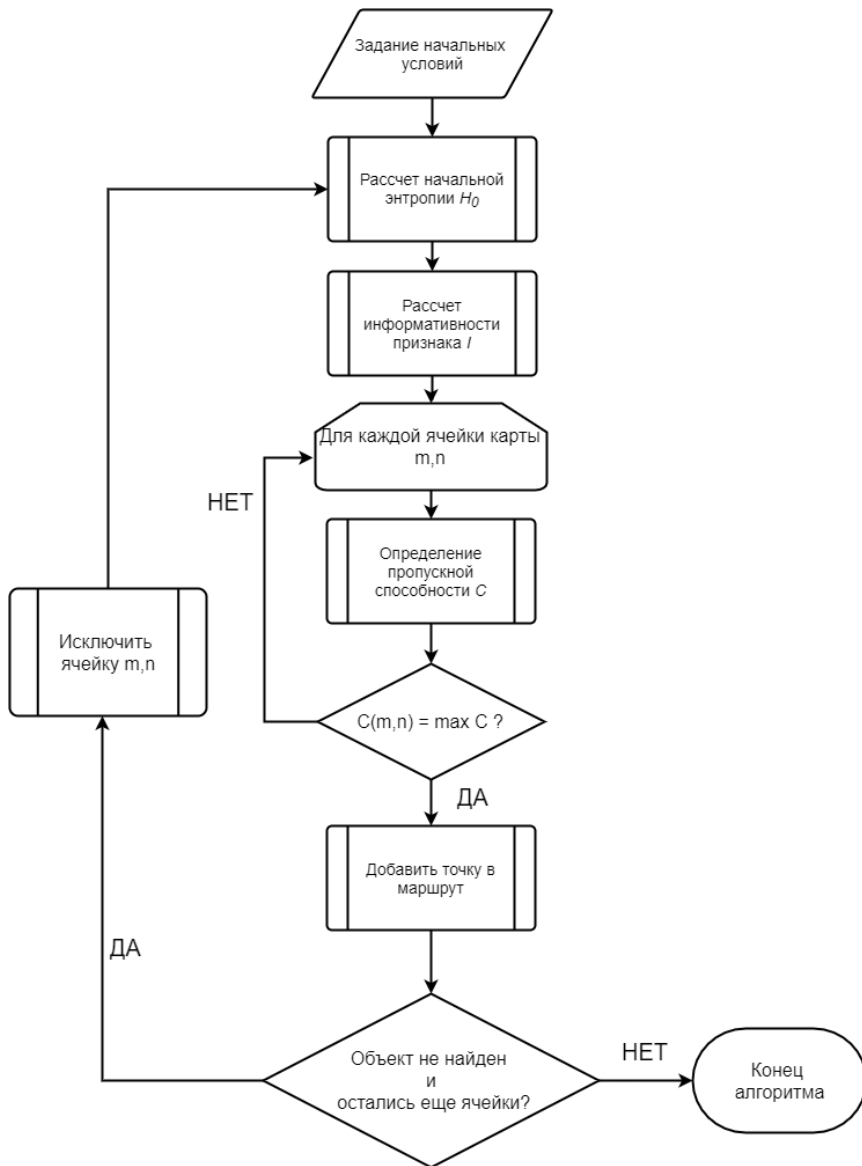


Рис. 1. Блок схема алгоритма построения маршрута

С учетом вышеописанных выражений алгоритм построения маршрута для одного БЛА будет следующим:

1. Задать начальные значения вероятностей нахождения ОИ и его отсутствия, условных плотностей распределения вероятности появления каждого признака для каждого класса в каждой ячейке карты;
2. Рассчитать начальную энтропию  $H_0$  по формуле (1);
3. В соответствии с выбранным поисковым признаком  $u_i$  рассчитать информативность каждой ячейки карты по формуле (5);
4. На каждом последующем шаге, с учетом информативности из п. 3, рассчитать пропускную способность.
5. Выбрать следующим пунктом маршрута ту ячейку, которая имеет максимальную пропускную способность.
6. Если в ячейке не находится ОИ, то эту ячейку исключить из дальнейшего поиска, путем обнуления априорной вероятности нахождения объекта в этой ячейке и перейти к п.2, иначе перейти к п. 7.
7. Работа алгоритма заканчивается если не осталось не обследованных ячеек или объект был найден.

### 3. Результаты математического моделирования

Для проверки работоспособности алгоритма было проведено математическое моделирование процесса поиска (на карте поиска размером  $M = 30, N = 30$ , состоящей из условных элементарных участков). Вероятность наличия ОИ  $P(o)$  представляет из себя сглаженную поверхность (рис. 2). Вероятность отсутствия ОИ определяется как  $P(b) = 1 - P(o)$ .

На борту БЛА имеется измерительное устройство, для которого известна плотность распределения вероятности приема некоторого значения признака, при условии наличия ОИ в зоне видимости измерителя. Данная плотность является гауссовской. Также известна плотность распределения вероятности приема некоторого значения, того же признака, при условии отсутствия объекта поиска в зоне видимости измерителя. Статистические характеристики условных плотностей (математическое ожидание и дисперсия) определены случайным образом, для каждой ячейки области поиска (рис.3):

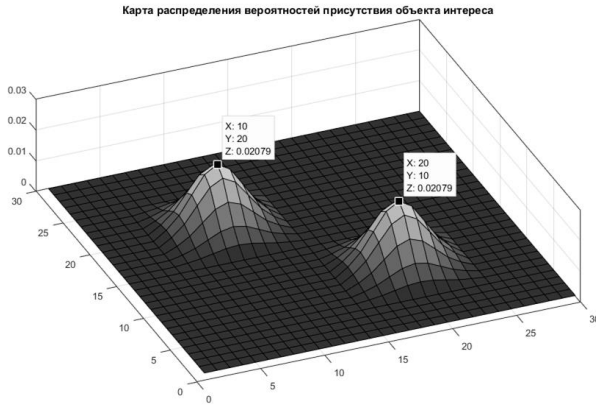


Рис. 2. Поверхность вероятностей наличия ОИ –  $P(o)$

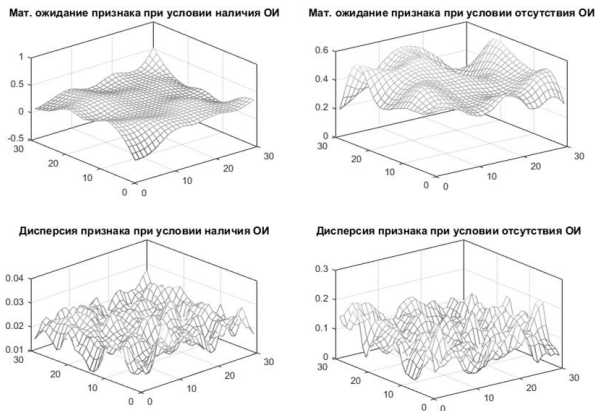


Рис. 3. Поверхности значений статистических характеристик условных плотностей вероятности приема значений признака

Из предложенного алгоритма следует, что БЛА должен выбирать те ячейки, которые обладают максимальной информативностью. Из рис. 2 видно, что в области поиска имеются две области, в которых вероятность нахождения ОИ выше. Это ячейки с координатами  $cell_1 = [x = 10; y = 20]$ ,  $cell_2 = [x = 20; y = 10]$ . Так как в этих ячейках одинаковые вероятности и расстояние от точки старта, то определяющим фактором будет наблюдаемость объекта поиска в этих ячейках.



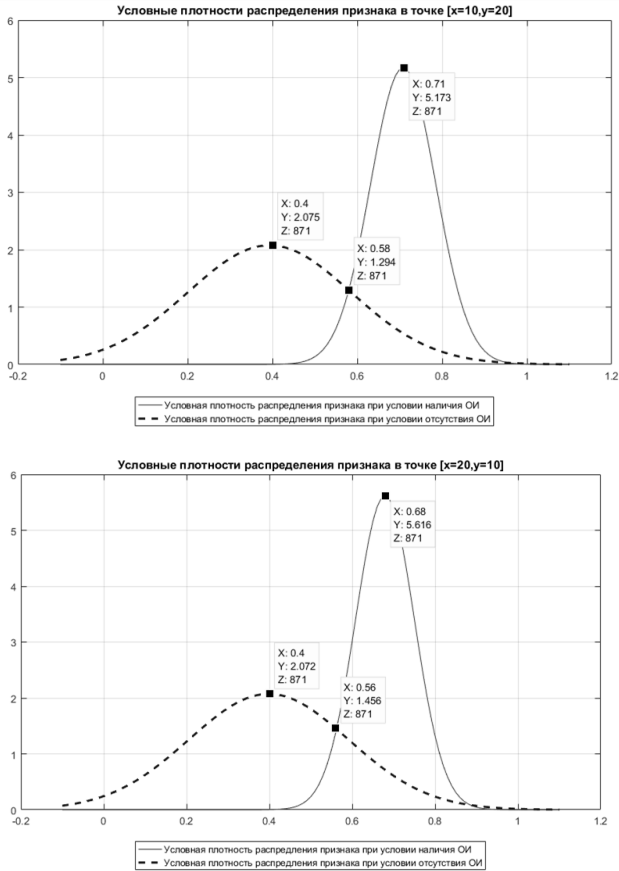


Рис. 4. Реализации условных плотностей распределения

На рис. 4 представлены реализации условных плотностей распределения, откуда видно, что плотности распределения для второй ячейки ( $cell_2$ ) имеют меньшую общую площадь пересечения, что говорит о меньшем количестве возможных ошибок и большей информативности данной ячейки.

На рис. 5 представлена поверхность распределения начальной информативности. Видно, что информативность второй ячейки больше.

Принято допущение о том, что БЛА движется с постоянной скоростью, условно равной  $v_{БЛА} = 1$ . Отсюда следует, что критерий пропускной

способности учитывает, только расстояние между текущей позицией маршрута и следующей информативной ячейкой.

Результаты работы алгоритма представлены на, рис. 6. ОИ располагался в ячейке с координатами  $[x = 12, y = 20]$ . Как видно из графиков БЛА решил задачу обнаружения ОИ за 71 итерацию.

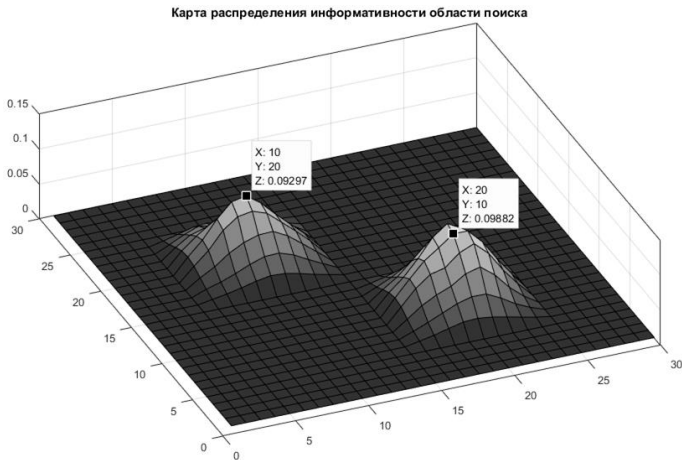


Рис. 5 Начальное распределение информативности

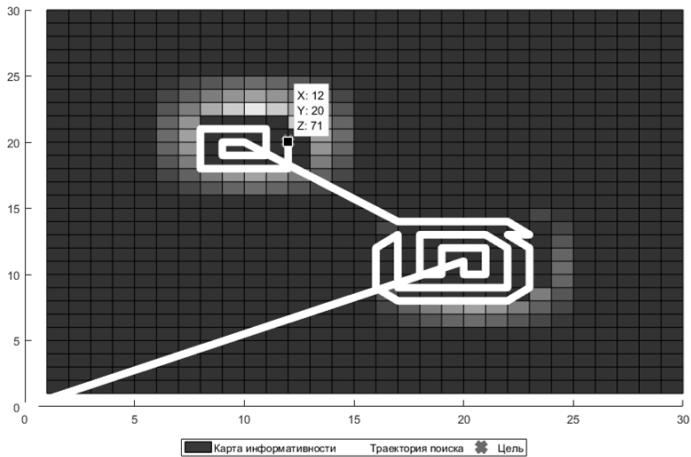


Рис. 6. Траектория поиска и траектория поиска на фоне поверхности распределения информативности

## Заключение

Компьютерное моделирование показало работоспособность предложенного алгоритма.

Основное преимущество данного подхода заключается в том, что при выборе очередной точки маршрута, учитывается не только вероятность присутствия ОИ, но и наблюдаемость конкретного признака, а также расстояние (которое может быть заменено на некоторую функцию затрат) до данной точки.

К недостаткам алгоритма можно отнести сложность определения условных плотностей распределения признака на реальных задачах.

Следующими шагами в исследовании данной проблемы будут:

- сравнительный анализ эффективности с классическими алгоритмами (регулярный поиск, случайный поиск);
- адаптация алгоритма к изменяющимся во времени вероятностям обнаружения ОИ и характеристикам условных плотностей распределения признаков.

## Список литературы

- [Абчук и др. 1977] Абчук, В.А., Суздаль, В.Г. "Поиск объектов", М.: Сов. радио; 1977. 336 с.
- [Андрейчук и др. 2016] А.А. Андрейчук, К.С. Яковлев "Метод разрешения конфликтов при планировании пространственных траекторий для группы беспилотных летательных аппаратов" // БТС-ИИ-2016, 22-23 сентября 2016, Иннополис, университет Иннополис. С. 31-40.
- [Ким 2001] Ким Н.В. "Обработка и анализ изображений в системах технического зрения": Учебное пособие. – М.: Изд-во МАИ, 2001 – 164 с.: ил.
- [Ким и др. 2002] Ким Н.В., Степанов Н.В. "Слежение за несколькими подвижными целями на основе информационного подхода"// Тез. 3-й научно-практ. конф. Современные проблемы фотограмметрии и дистанционного зондирования. – 2002.
- [Коган 1981] Коган И.М. "Прикладная теория информации" – М.: Радио и связь. 1981. —216 с.
- [Лавренов и др. 2016] Р.О. Лавренов, И.М. Афанасьев, Е.А. Магид "Планирование маршрута для беспилотного наземного робота с учетом множества критериев оптимизации" // БТС-ИИ-2016, 22-23 сентября 2016, Иннополис, университет Иннополис. С. 10-20.

УДК 681.51

## ДВИЖЕНИЕ ГРУППЫ МОБИЛЬНЫХ РОБОТОВ В СТРОЮ ТИПА “КОНВОЙ” — ТЕОРИЯ, МОДЕЛИРОВАНИЕ И ЭКСПЕРИМЕНТ

С.Л. Зенкевич (*zenkev@bmstu.ru*)Хуа Чжу (*zhuhua1302@gmail.com*)Цзяньвень Хо (*huojianwen2008@hotmail.com*)

Научно-учебный центр «робототехника» МГТУ им. Н.Э.

Баумана, Москва

**Аннотация.** В статье решен ряд задач, связанных с движением группы мобильных роботов в строю типа “конвой”. Описаны способ локализации ведомых роботов и метод управления ими. Указано разумное количество роботов в таком типе движения. Проведено моделирование движения группы роботов в среде Stage (+ROS). Проведен эксперимент с участием реальных мобильных роботов. Разработана распределенная система управления ведущими и ведомыми роботами в среде ROS. Приведены результаты экспериментов, в том числе, построенная карта среды ведущим роботом и траектории роботов в группе.

**Ключевые слова:** группа мобильных роботов, конвой, управление, локализация, эксперимент, ROS.

### Введение

Часто возникает ситуации, когда один мобильный робот не в состоянии выполнить задачу и возникает проблема согласованного управления группой роботов. Одной из важных задачей такого класса является согласованное движение группы роботов, образующих строй типа «конвой».

Приложения, связанные с такого типа движением, состоят, например, в смене дислокации группы роботов, в перевозке большого количества грузов, которые каждый робот в отдельности не в состоянии перевезти. В этой ситуации ведущий робот должен быть оснащен богатой сенсорной системой, позволяющей планировать траекторию в непредсказуемой обстановке, например, сканирующими лазерными дальномерами, системами спутниковой навигации, стереозрением. Ведомые роботы в этом случае должны только следить за положением своего ведущего робота.

Осуществлению движения группы роботов в строю типа “конвой” в настоящее время уделяется большое внимание [Das et al., 2002] [Chiem et al., 2004] [Thuilot et al., 2004] [Yazbeck et al., 2013] [Yazbeck et al., 2014] [Mark, 2014]. В таблице 1 представлены сравнительные характеристики существующих методов.

Таблица 1

Сравнение существующих методов

Метод	Принцип работы или цель управления	Преимущество	Недостаток
1	Решение задачи следования или стабилизация	Простота реализации	Возможность столкновения с препятствиями; низкая точность
2	Аппроксимация траектории кривой Безье	Простота реализации	Зависимость точности следования от точности построения кривой Безье
3	Следование к ближайшей точке	Высокая точность	Возможность возникновения колебаний
4	Максимальная степень совпадения траектории роботов	Высокая точность	Высокая вычислительная сложность

## 1. Модель роботов

Пусть мобильные роботы движутся в плоскости  $OXY$  строем типа “конвой” (рис. 1). Роботы пронумерованы от  $R_0$  – ведущего робота до  $R_N$  – замыкающего конвой. Этот тип движения требует, чтобы каждый  $i$  –ый робот,  $i = 1, \dots, N$ , кроме ведущего, двигался по траектории, вдоль которой движется  $i - 1$  –ый робот, с некоторым временным запаздыванием  $T$ .

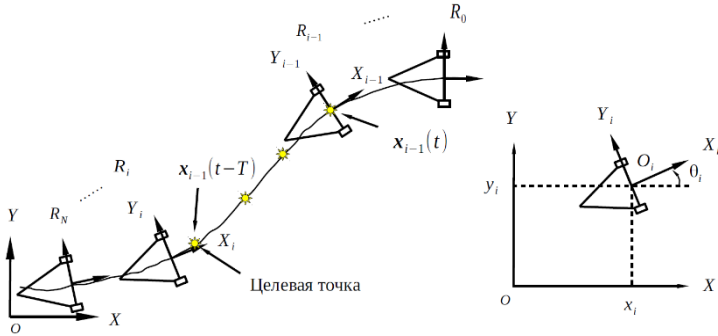


Рис. 1. Строй типа «конвой»

Ведущий робот  $R_0$  либо управляется оператором, либо является полностью автономным. В том случае, когда ведущий робот автономен, он оснащен спутниковой системой навигации или сканирующим лазерным дальномером. Остальные роботы имеют систему одометрии и менее мощные лазерные сканеры. Кроме того, все роботы являются узлами беспроводной локальной сети, следовательно, могут обмениваться информацией друг с другом.

Математическая модель роботов имеет вид [Зенкевич и др., 2006]:

$$\begin{pmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{pmatrix} = \begin{pmatrix} \cos\theta_i & 0 \\ \sin\theta_i & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_i \\ \omega_i \end{pmatrix} = J(\theta_i) \begin{pmatrix} v_i \\ \omega_i \end{pmatrix} \quad (1)$$

где  $x_i$   $y_i$   $\theta_i$  – положение и ориентация робота,  $\omega_i$  – угловая скорость робота, а  $v_i$  – линейная скорость начала связанной системы координат  $O_i$  (далее линейная скорость робота).

## 1.1 Управление роботами

Стратегия управления для каждого ведомого робота состоит в выполнении соотношения:

$$\mathbf{x}_i(t) = \mathbf{x}_{i-1}(t - T) \quad (2)$$

где,  $\mathbf{x}_i(t)$  представляет собой положение робота  $R_i$  в текущий момент,  $\mathbf{x}_{i-1}(t - T)$  — положение робота  $R_{i-1}$  в момент времени  $t - T$ .

Заметим, что такая стратегия предлагалась одним из авторов в 1973-1976 гг. в работах, связанных с исследованием управления движением самолётов в строю типа конвой. Преимущество такого метода состоит в том, во-первых, расстояние между роботами является переменной, т.е. чем быстрее скорости движения конвоя, тем больше расстояние между роботами, а параметр управления  $T$  является постоянным; во-вторых, при проходе через

узкую область ведомый робот успешно обходит препятствие, который уже обходит его ведущий робот.

Для реализации стратегии необходимо:

1. Запоминание фрагмента траектории ведущего робота

В процессе движения каждый ведомый робот  $R_i$  имеет в качестве программной траектории сдвинутую во времени траекторию ведущего робота  $R_{i-1}$ , динамически меняющуюся во времени. Характер этой траектории, учитывающий дискретность измерений, представлен на рис. 2, где  $N$  — число точек,  $T/N$  — скважность измерений.

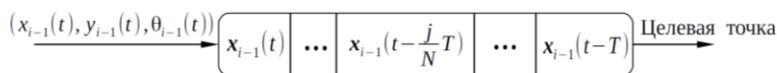


Рис. 2. Процесс нахождения целевых точек для робота  $R_i$

2. Формирование закона управления движением робота вдоль траектории

Закон управления состоит в способе движения робота вдоль динамически меняющейся траектории, заданной набором точек (рис. 2), и представляющей очередь.

Будем считать, что время обновления данных  $T/N$  мало по сравнению с инерционными свойствами робота. Это означает, что на вход системы управления поступает непрерывная последовательность в виде программной траектории  $(x^*(t), y^*(t), \theta^*(t))$ . Тогда из уравнения (1) следует соотношение:

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = J^+(\theta) \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \dot{x} \cos \theta + \dot{y} \sin \theta \\ \dot{\theta} \end{pmatrix} \quad (3)$$

где,  $J^+(\theta)$  — псевдообратная матрица,  $J^+(\theta) = (J^T J)^{-1} J^T$ , а линейная скорость равна проекции вектора  $(\dot{x}, \dot{y})$  на направление движения робота.

Принимая:

$$\begin{aligned} \dot{x} &= \frac{\Delta x}{\Delta t} = \frac{1}{\Delta t} (x^* - x), \\ \dot{y} &= \frac{\Delta y}{\Delta t} = \frac{1}{\Delta t} (y^* - y), \\ \dot{\theta} &= \frac{\Delta \theta}{\Delta t} = \frac{1}{\Delta t} (\theta^* - \theta), \end{aligned}$$

Окончательно имеем закон управления в виде:

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} k_1(x^* - x) \cos \theta + k_1(y^* - y) \sin \theta \\ k_2(\theta^* - \theta) \end{pmatrix} \quad (4)$$

где  $k_1, k_2$  — параметры.

На рис. 3 представлена схема, иллюстрирующая формирование управляющих сигналов, обеспечивающих движение робота вдоль произвольной траектории.

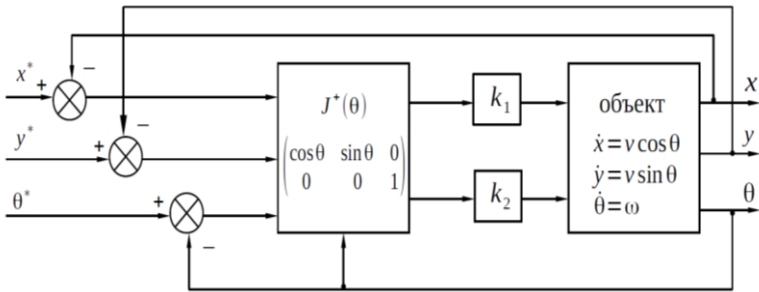


Рис. 3. Структурная схема управления роботом

## 1.2. Локализация ведомых роботов

Предложенный способ локализации роботов состоит в том, что каждый робот определяет собственное положение, используя взаимное положение роботов, при которых необходимо знание глобального положения ведущего робота. Для осуществления такого способа требуется выполнение трёх условий:

- знание глобального положения ведущего робота;
- знание положения ведущего робота относительно ведомого;
- возможность передавать информацию о положении ведомому роботу

Используя эту информацию, каждый робот определяет собственное положение в абсолютной системе координат. В процессе решения задачи управления движением каждого робота строя возникает проблема определения собственного положения каждого робота  $R_{i-1}$  относительно  $R_i$ . В лабораторных условиях эта проблема может быть решена путем установки систем зрения, определяющих декартовы координаты всех членов конвоя, что является невозможным при движении роботов в открытом пространстве.

В статье [Зенкевич и др., 2017] используется известный расширенный фильтр Калмана для обработки измерений, проводимых лазерным сканером. На рис. 4 показана зависимость погрешности локализации ведомых роботов в строю типа “конвой” от номера робота  $R_i$  ( $i = 1, 2, \dots, 10$ ).

Из результатов моделирования можно сделать следующие выводы:

- Из-за неизбежных неточностей измерений, полученных от лазерного сканера, погрешности накапливаются, что естественно.



2. Как видно из рис. 4, погрешности локализации ведомых роботов с 6 по 10 увеличились с течением времени, в то же время, погрешность локализации роботов с 1 по 4 стали почти постоянными с течением времени.

3. С точки зрения локализации с помощью используемого метода, наилучшим количеством роботов в строю типа “конвой” является 5, т.е., один ведущий робот и четыре ведомых роботов.

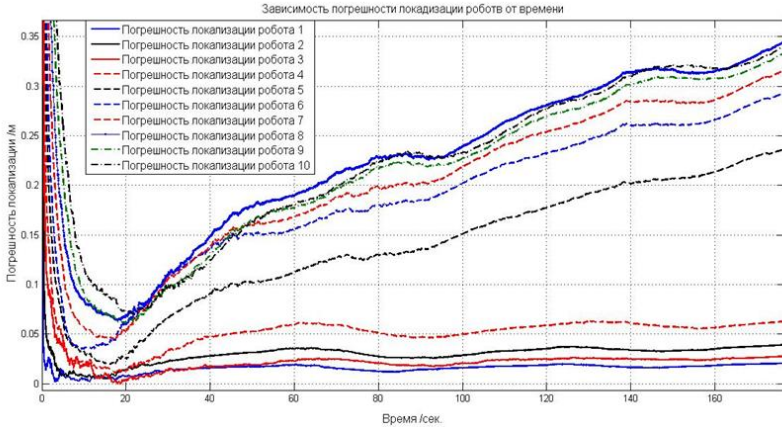


Рис. 4. Погрешности локализации ведомых роботов

## 2. Практическая часть

### 2.1. Моделирование движения роботов в группе

В настоящее время существуют различные программные среды для моделирования движения группы роботов, в том числе, ARGoS (Autonomous Robots Go Swarming), Gazebo/Stage. В настоящей работе для моделирования двумерного движения роботов используется эмулятор Stage\_gos в среде ROS. Моделируется движение “конвой” из одиннадцати роботов. Ведущий робот  $R_0$  совершает движение с постоянной линейной скоростью  $v_i = 5 \text{ m/s}$  и угловой скоростью  $\omega_i = 0.1 \text{ rad/s}$  при  $t < 62.8 \text{ s}$ ,  $\omega_i = -0.1 \text{ rad/s}$  при  $t > 62.8 \text{ s}$ , т.е. движется по траектории типа «восьмерка». Скорости остальных роботов вычисляются в соответствии с (5):

$$\begin{pmatrix} v_i \\ \omega_i \end{pmatrix} = \begin{pmatrix} k_1(x_{i-1}(t-T) - x_i(t))\cos\theta_i(t) + k_1(y_{i-1}(t-T) - y_i(t))\sin\theta_i(t) \\ k_2(\theta_{i-1}(t-T) - \theta_i(t)) \end{pmatrix} \quad (5)$$

Как видно из результатов моделирования, траектории роботов в группе почти совпадают, а также движение последнего робота не имеет колебаний (видно из его скорости).

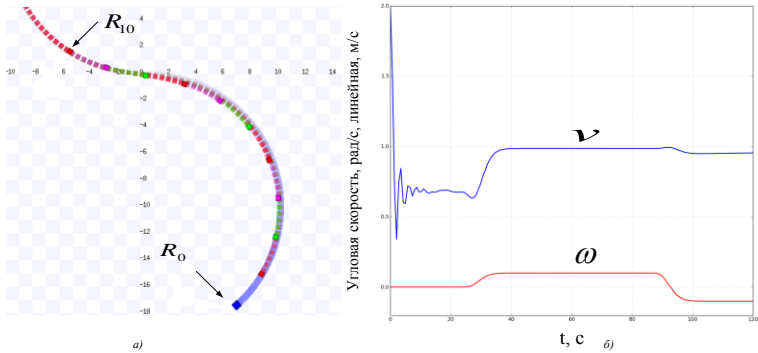
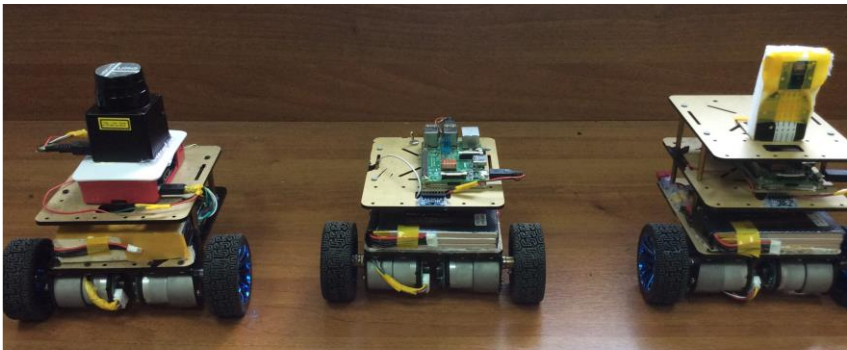


Рис. 4. Результаты моделирования, а) Траектории роботов б) Скорость последнего робота  $R_{10}$

## 2.2. Экспериментальное исследование

Для дальнейшей проверки работоспособности предложенного алгоритма, проведен эксперимент для осуществления этого типа движения с участием реальных мобильных роботов, внешний вид которых представлен на рис. 5, их структурная схема показана на рис. 6.



а) Ведущий робот

б) Ведомый робот

в) Последний ведомый робот

Рис. 5. Внешний вид роботов

В состав робота входят следующие компоненты: 1). Механическая платформа, имеющая два независимых ведущих колеса и одно пассивное.

2). Два электродвигатель с магнитными энкодерами. 3). Одноплатный компьютер Raspberry Pi 2B (частота: 900MHz, ОЗУ: 1гб). 4). Микроконтроллер Arduino Uno для управления двигателями. 5). Лазерный сканер Hokuyo urg-04lx-urg01 (диапазон измерения: 5.6 м, скорость сканирования: 100мс/скан, только у ведущего робота). 6). Raspberry Pi камера (только у последнего ведомого робота).

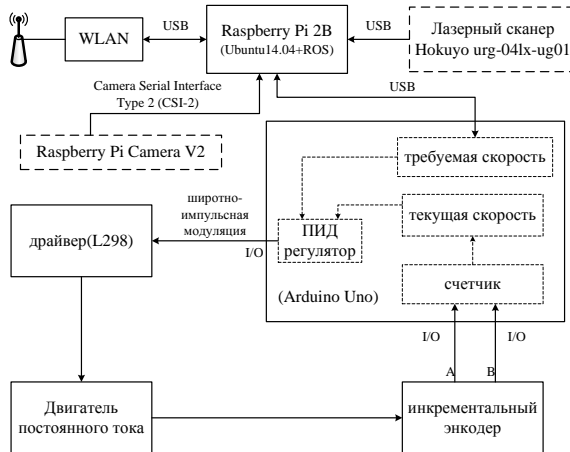


Рис. 6. Структурная схема робота

Цель эксперимента состоит в том, чтобы в условиях отсутствия априорных знаний о внешней среде перевести группу мобильных роботов из начального положения в положение, заданное оператором. Для реализации такой системы в масштабе реального времени, роботы оснащены операционной системой Ubuntu 14.04, совместно с ROS (Robot Operating System). В среде ROS построена распределённая система управления роботами (рис. 7.).

Процесс `/hector_mapping` используется для построения двухмерной карты окружающей среды и определения собственного положения в ней с помощью информации лазерного сканера, обслуживаемого процессом `/hokuyo_node`. Процесс `/move_base` служит для динамического формирования роботом отслеживаемой траектории, а также формирования команды управления. Процесс `/follower` ведомого робота предназначен для отслеживания им траектории своего ведущего робота. Определение положения ведомого робота в окружающей среде осуществляется процессом `/odometry`. Управление исполнительном механизмом осуществляется процессом `/serial_node`. Процесс `/raspicam_node`, принад-

лежащий последнему ведомому роботу, служит для мониторинга движения роботов оператором с помощью камеры, установленной на роботе. Из-за ограниченной вычислительной мощности бортового компьютера, части задачи ведущего робота выполняются удаленным компьютером.

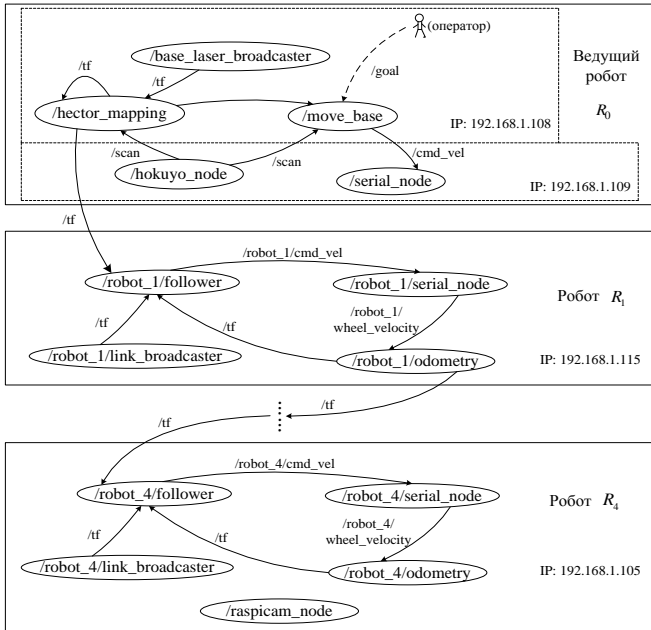


Рис. 7. Распределённая система управления роботами

Были проведены два эксперимента

*а) Проверки функционирования в автономном режиме ведущего робота*

При заданной оператором последовательности целевых точек А, В, С, D, ведущий робот автономно проходил эти целевые точки, в процессе движения распознавая окружающую среду и планируя траекторию. Реальная траектория робота показана на рис. 8.

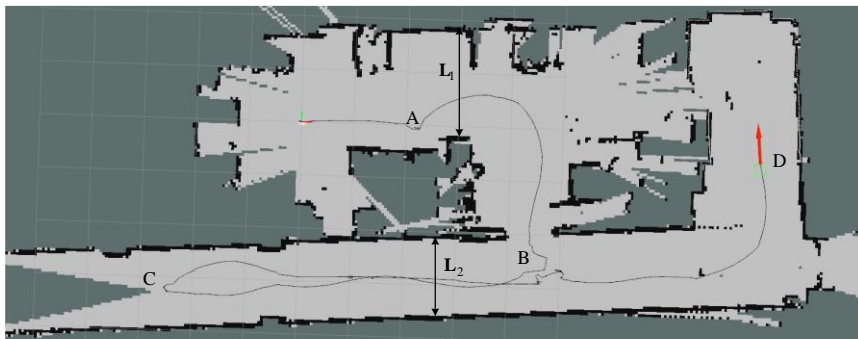


Рис. 8. Траектория ведущего робота в построенной карте

Точности построенной карты может характеризоваться таблицей 2.

Таблица 2

Сравнение реальной дистанции и дистанции на карте

Имя отрезки	Реальная дистанция (m)	Дистанция на карте (m)	Погрешность (m)	%
$L_1$	2.05	1.98539	0.06461	3.15
$L_2$	1.5	1.43363	0.06637	4.4

*б) Проверки работоспособности закона управления движения группы роботов в строю типа “конвой”*

При движении робота в строю линейная скорость ведущего робота ограничена в диапазоне  $\left[0.1 \frac{m}{s}, 0.2 \frac{m}{s}\right]$ , угловая скорость ограничена в диапазоне  $\left[-1.5 \frac{rad}{s}, 2 \frac{rad}{s}\right]$ . Безопасное расстояние между роботами установлено 0.15 m. Результаты движения показаны на рис. 9.

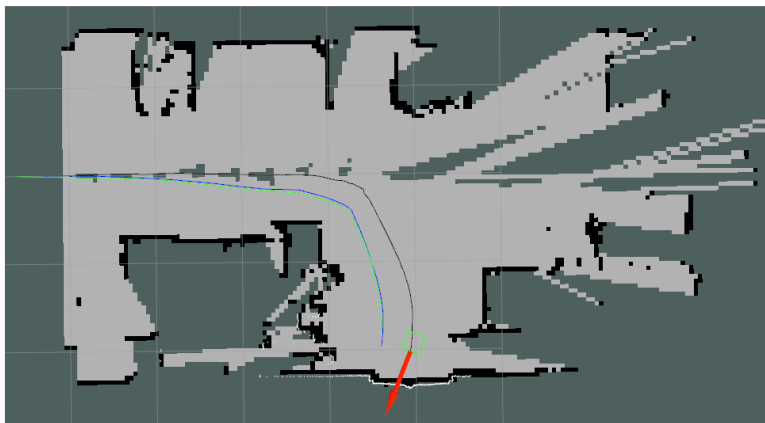


Рис. 9. Траектории роботов  $R_0$  (чёрная),  $R_1$  (синяя) и  $R_2$  (жёлтая)

## Заключение

Исследованы движения группы мобильных роботов в строю типа “конвой”. Разработаны закон управления движением ведомых роботов и их способы локализации. Получено разумное количество роботов в таком типе движения по результатам моделирования способы локализации. Доказана работоспособность предложенного метода управления роботами по результатам моделирования и эксперимента.

Отметим, что в данной работе рассматривается только один случай задачи конвоирования, другие случаи задачи конвоирования, например, вход робота в конвой, выход робота из конвоя, объединение конвоев, разделение конвоя и т.д., т.е. изменения топологии строя, рассматриваются в следующей работе авторов «логическое управление группой мобильных роботов с помощью конечного автомата».

## Список литературы

- [Das et al., 2002] Das K., Fierro R., Kumar V. Ostrowski J. P., Spletzer J., Taylor C. J. A vision-based formation control framework // IEEE Transactions on Robotics and Automation. — 2002. — Vol. 18, № 5. — P. 813-825.
- [Chiem et al., 2004] Chiem S., Cervera E. Vision-based robot formations with Bezier trajectories // In: Proceedings of the 8th Conference on Intelligent and Autonomous System. — 2004. — P. 191-198.
- [Yazbeck et al., 2013] Yazbeck J., Scheuer A., Charpillat F. Optimized lateral control for a decentralized near-to-near platooning. // INRIA, Tech. Rep. — 2013.

- [**Thuilot et al., 2004**] Thuilot B., Bom J., Marmoiton F., Martinet P. Accurate automatic guidance of an urban electric vehicle relying on a kinematic GPS sensor. // In: 5th IFAC Symposium on Intelligent Autonomous Vehicles, Lisboa (Portugal). — 2004.
- [**Yazbeck et al., 2014**] Yazbeck J., Scheuer A., Charpillat F. Decentralized Near-to-Near Approach for Vehicle Platooning based on Memorization and Heuristic Search. // IEEE International Conference on Robotics & Automation. — 2014. — P. 631-638
- [**Mark, 2014**] Mark Daniel Ollis. Position estimation and vehicle control in autonomous multi-vehicle convoys. // Patent Application Publication, Pub. №: US 2014/0309836 A1.
- [**Зенкевич и др., 2006**] Зенкевич С. Л., Назарова А. В. Система управления мобильного колесного робота // Вестник МГТУ им. Н. Э. Баумана. Сер. “приборостроение”. 2006. №3. С. 31-51.
- [**Зенкевич и др., 2017**] Зенкевич С. Л., Чжу Хуа Управление движением группы роботов в строю типа “конвой” // Мехатроника, автоматизация, управление. 2017. Т.18, №1. С. 30-34.
- [**Чжу, 2014**] Чжу Хуа. Система управления коллективом мобильных роботов. // Молодежный научно-технический вестник. 2014. № 12. С. 48.

УДК 519.71, 531.32, 62-529, 629, 623.1/7:623.41

## ПЛАНИРОВАНИЕ ЗАКОНОВ ДВИЖЕНИЯ НАЗЕМНЫМ БТС НА ОСНОВЕ ЦЕЛЕВОЙ ФУНКЦИИ, ЗАДАВАЕМОЙ ОПЕРАТОРОМ

А.В. Леонард (*alex-leonard@yandex.ru*)

Е.С. Брискин (*dtm@vstu.ru*)

И.А. Горбов (*gorbovigor34@yandex.ru*)

Волгоградский государственный технический  
университет, Волгоград

**Аннотация.** В статье рассматривается вопрос разработки алгоритма для синтеза законов движения системой планирования беспилотного транспортного средства (БТС) при обходе препятствия. Синтез законов движения базируется на использовании интегрального критерия – целевой функции в виде линейной свертки частных показателей с весовыми коэффициентами, которая отражает проектируемую специфику работы БТС.<sup>1</sup>

**Ключевые слова:** локальное планирование движения БТС, вариационное исчисление, база знаний робота, недетерминированная среда

### Введение

При планировании своего движения, робот должен:

1. Иметь возможность получать от оператора *цель своего движения*;
2. Быть способным *оценить состояние окружающей среды*, в которой он находится, что достигается за счет использования необходимого числа сенсоров (датчиков) [Юревич и др., 2007], интегрированных в систему управления;
3. В соответствии с целью своего движения и состоянием окружающей среды самостоятельно осуществить *планирование движения* на основе, заложенной в его бортовой или удаленной части системы управления, базы знаний (алгоритмов) [Михайлов и др., 2016].

---

<sup>1</sup> Работа выполнена при финансовой поддержке РФФИ (проект № 16-38-60086 мол\_a\_дк, № 17-01-00675).



В этом смысле можно говорить об интеллектуализации транспортного средства, как о способности им самостоятельно перемещаться [Lin Rui et. al., 2009], [Darintsev et. al., 2007], [Hiroshi Koyasu et. al., 2006], [Eiji Uchibe et. al., 2000] в не детерминированной среде.

На сегодняшний день исследования по *планированию движения* роботов ведутся многими научными коллективами. Предлагаются различные методы: на основе графов видимости, диаграмм Вронского, неинформированный поиск, эвристический поиск, Алгоритм Lifelong Planning A\*, Visibility PRM, Vertex Enhancement, Нить Ариадны, Rapidly-Exploring random trees, RRT и др. Но аналитический обзор литературных источников порядка 300-сот наименований [Казаков и др., 2016] (в том числе по запросам в отечественной наукометрической базе elibrary) показывает, что предлагаемые методы ориентированы на синтез пути следования робота в детерминированной (не детерминированной) среде, а вопрос реализации этих траекторий по существу не затрагивается. Иными словами в этих методах (сэмплирования, дискретизации пространства сцены) задача планирования движения в лучшем случае сводится к постановке оптимизационной задачи с целевой функцией, но эта функция во многих работах соответствует минимальной длине маршрута и (или) максимальной удаленности перемещаемого объекта от препятствий. А формирующиеся траектории, на основе этих методов, за исключением методов гравитирующих сил (потенциальных полей) [Khatib, 1986], представляют собой кусочные функции с разрывами первого рода. Что на практике, даже если законы движения будут сформированы другими подходами, - проблематично реализуемо [Мартыненко, 2005] или потребует дополнительных мер по сглаживанию маршрута [Лавренов и др., 2016]. Поэтому задача комплексного планирования движения: совместного формирования законов движения и изначально гладкой траектории, огибающей препятствия, является актуальной задачей и обладает определенной новизной.

В данной работе авторы *ставят задачу: разработать метод оптимального планирования законов движения БТС для обхода протяженных статичных препятствий в заранее недетерминированной среде, если робот моделируется материальной точкой на плоскости.*

## 1. Теоретическая проработка метода

### *Формирование начальной базы знания БТС. Действия оператора:*

1. Выбирает одну из возможных целевых функций [Васильев, 2012], [Жога и др., 2010], [Statnikov et. al., 1995] - критерий, отражающий будущую специфику поведения робота при обходе препятствия из пункта старта в пункт назначения:

$$I = k_1 \int_0^{\tau} (\dot{x}^2 + \dot{y}^2) dt - k_2 \int_0^{\tau} (\sqrt{x^2 + y^2} - R) dt + k_3 \int_0^{\tau} (\ddot{x}^2 + \ddot{y}^2) dt \rightarrow \min \quad (1)$$

где  $x, y$  – координаты;  $\dot{x}, \dot{y}, \ddot{x}, \ddot{y}$  – линейные скорости и ускорения в декартовой системе координат центра масс (характерной точки корпуса);  $R$  – расстояние от начала системы координат до границы препятствия, моделируемого на плоскости окружностью;  $\tau$  – времени следования из пункта старта в пункт назначения характерной точки корпуса;  $k_{1(2,3)}$  – весовые коэффициенты, отражающие значимость частных показателей.

По критерию (1) робот должен быть способен минимизировать квадрат скорости, квадрат ускорения и максимизировать расстояние до препятствия [Казаков и др., 2016], [Лавренов и др., 2016] в каждый момент времени синтезируемого движения.

Задается условиями экстремума соответствующего порядка в виде уравнений Эйлера – Пуассона [Охоцимский и др., 1957], [Брискин и др., 2011], [Briskin et. al., 2014a], [Ловейкин и др., 2014]:

$$\begin{cases} \frac{\partial \Phi}{\partial x} - \frac{d}{dt} \left( \frac{\partial \Phi}{\partial \dot{x}} \right) + \frac{d^2}{dt^2} \left( \frac{\partial \Phi}{\partial \ddot{x}} \right) = 0, \\ \frac{\partial \Phi}{\partial y} - \frac{d}{dt} \left( \frac{\partial \Phi}{\partial \dot{y}} \right) + \frac{d^2}{dt^2} \left( \frac{\partial \Phi}{\partial \ddot{y}} \right) = 0, \end{cases} \quad (2)$$

где  $\Phi = k_1 (\dot{x}^2 + \dot{y}^2) - k_2 (\sqrt{x^2 + y^2} - R) + k_3 (\ddot{x}^2 + \ddot{y}^2)$ .

Осуществляет преобразования:

$$\begin{aligned} \frac{\partial \Phi}{\partial x} &= -k_2 (x^2 + y^2)^{-0.5} x; & \frac{\partial \Phi}{\partial y} &= -k_2 (x^2 + y^2)^{-0.5} y; \\ \frac{\partial \Phi}{\partial \dot{x}} &= 2k_1 \dot{x}; & \frac{\partial \Phi}{\partial \dot{y}} &= 2k_1 \dot{y}; & \frac{\partial \Phi}{\partial \ddot{x}} &= 2k_3 \ddot{x}; & \frac{\partial \Phi}{\partial \ddot{y}} &= 2k_3 \ddot{y}; \\ \frac{d}{dt} \left( \frac{\partial \Phi}{\partial \dot{x}} \right) &= 2k_1 \ddot{x}; & \frac{d}{dt} \left( \frac{\partial \Phi}{\partial \dot{y}} \right) &= 2k_1 \ddot{y}; & & & & \\ \frac{d^2}{dt^2} \left( \frac{\partial \Phi}{\partial \ddot{x}} \right) &= 2k_3 x^{IV}; & \frac{d^2}{dt^2} \left( \frac{\partial \Phi}{\partial \ddot{y}} \right) &= 2k_3 y^{IV}, & & & & \end{aligned} \quad (3)$$

откуда определяются оптимальные законы движения в дифференциальной форме:

$$\left\{ \begin{aligned} x^{IV} &= \frac{k_1}{k_3} \ddot{x} + \frac{k_2}{2k_3} (x^2 + y^2)^{-0.5} x; & y^{IV} &= \frac{k_1}{k_3} \ddot{y} + \frac{k_2}{2k_3} (x^2 + y^2)^{-0.5} y. \end{aligned} \right. \quad (4)$$

Программирует дифференциальные уравнения (4) и методы их численного интегрирования или расчета постоянных интегрирования для

аналитической формы в системе самоуправления робота (базе знаний робота);

Задаёт возможность ввода начальных или граничных условий по четыре для каждого уравнения (4) из 16 возможных:  $x_0, y_0, \dot{x}_0, \dot{y}_0, \ddot{x}_0, \ddot{y}_0, \ddot{\ddot{x}}_0, \ddot{\ddot{y}}_0, x_\tau, y_\tau, \dot{x}_\tau, \dot{y}_\tau, \ddot{x}_\tau, \ddot{y}_\tau, \ddot{\ddot{x}}_\tau, \ddot{\ddot{y}}_\tau$  и время  $\tau$ , тем самым определяя роботу *цель его движения* в недетерминированных условиях;

Кодирует алгоритм определения параметра  $R$  в зависимости от значений расстояний до поверхности препятствия, поступающих от лазерных дальномеров;

Например, для цели движения – перемещения из пункта расположения в пункт назначения:  $\tau, x_0, y_0, \dot{x}_0, \dot{y}_0, x_\tau, y_\tau, \dot{x}_\tau, \dot{y}_\tau$ , программирует метод выбора коэффициентов  $k_{1(2,3)}$  и определения начальных параметров  $\ddot{x}_0, \ddot{y}_0, \ddot{\ddot{x}}_0, \ddot{\ddot{y}}_0$ , необходимых для интегрирования уравнений (4), траектория которых не пересекает обнаруженное препятствие.

В результате предложенной последовательности действий оператора осуществляется перенос функций планирования движения в систему управления робота. Робот становится способным планировать свое движение самостоятельно.

*Действия БТС в соответствии с запрограммированной базой знания:*

В процессе движения БТС лазерные дальномеры с заданным интервалом времени осуществляют поиск препятствия перед роботом.

1. По заданной оператором системе уравнений (4), соответствующих целевой функции (1) и *цели движения*, например, - переместится из пункта А в пункт В:  $x_0, y_0, \dot{x}_0, \dot{y}_0, x_\tau, y_\tau, \dot{x}_\tau, \dot{y}_\tau$ , за время  $\tau$ , осуществляет построение законов движения. Если препятствие до начала движения не обнаружено, то в дифференциальных уравнениях (4):  $k_2 = 0$ , тогда

$$\left\{ x^{IV} = \frac{k_1}{k_3} \ddot{x}; y^{IV} = \frac{k_1}{k_3} \ddot{y} \Leftrightarrow \begin{cases} x = A_1 + A_2 t + A_3 e^{\sqrt{\frac{k_1}{k_3}} t} + A_4 e^{-\sqrt{\frac{k_1}{k_3}} t}; \\ y = B_1 + B_2 t + B_3 e^{\sqrt{\frac{k_1}{k_3}} t} + B_4 e^{-\sqrt{\frac{k_1}{k_3}} t} \end{cases} \quad (5)$$

В соответствии с целью движения система управления БТС должна рассчитать  $A_j, B_j$  согласно следующим системам уравнений

$$\left\{ \begin{array}{l} x_0 = A_1 + A_3 + A_4; \\ x_\tau = A_1 + A_2\tau + A_3e^{\sqrt{\frac{k_1}{k_3}}\tau} + A_4e^{-\sqrt{\frac{k_1}{k_3}}\tau}; \\ \dot{x}_0 = A_2 + A_3\sqrt{\frac{k_1}{k_3}} - A_4\sqrt{\frac{k_1}{k_3}}; \\ \dot{x}_\tau = A_2 + A_3\sqrt{\frac{k_1}{k_3}}e^{\sqrt{\frac{k_1}{k_3}}\tau} - \sqrt{\frac{k_1}{k_3}}A_4e^{-\sqrt{\frac{k_1}{k_3}}\tau}; \end{array} \right. \quad \left\{ \begin{array}{l} y_0 = B_1 + B_3 + B_4; \\ y_\tau = B_1 + B_2\tau + B_3e^{\sqrt{\frac{k_1}{k_3}}\tau} + B_4e^{-\sqrt{\frac{k_1}{k_3}}\tau}; \\ \dot{y}_0 = B_2 + B_3\sqrt{\frac{k_1}{k_3}} - B_4\sqrt{\frac{k_1}{k_3}}; \\ \dot{y}_\tau = B_2 + B_3\sqrt{\frac{k_1}{k_3}}e^{\sqrt{\frac{k_1}{k_3}}\tau} - \sqrt{\frac{k_1}{k_3}}B_4e^{-\sqrt{\frac{k_1}{k_3}}\tau}; \end{array} \right.$$

одним из методов решения системы алгебраических уравнений (например, методом Гаусса). Для более сложных критериев необходим численный поиск неизвестных констант интегрирования с использованием заданных граничных условий. Построение законов осуществляется в неподвижной системе координат, связанной с начальным положением характерной точки корпуса БТС.

2. Запускает программу практической реализации синтезированных законов движения (5) и их стабилизации.

3. В случае, если препятствие обнаружено, происходит перепланирование движения в соответствии с системой (4) ( $k_2 \neq 0$ ), для которой определяются свои граничные условия из системы (5) путем подстановки в нее  $\tau_1$  - времени обнаружения препятствия, или путем опроса датчиков; При этом осуществляется перерасчет координат, входящих в граничные условия, так как начало неподвижной системы координат смещается в центр окружности, моделирующей препятствие. Запускается численный метод поиска  $\ddot{x}_0, \ddot{y}_0, \ddot{x}_0, \ddot{y}_0$ , включающий в себя численное интегрирование уравнений (4) на интервале  $0 \leq t \leq \tau - \tau_1$  и проверку отсутствия пересечения синтезируемой траекторией контуров препятствия

$$\sqrt{x^2 + y^2} - R > 0 \quad (6)$$

на каждом шаге интегрирования.

Пример численного синтеза законов движения БТС в случае изначального обнаружения препятствия проиллюстрирован на Рис. 1. Где на Рис. 1 а) представлена траектория центра масс БТС для обхода препятствия, моделируемого окружностью. Рис. 1 б) и в) отражает изменение линейных скоростей и ускорений центра масс робота во времени в проекции на неподвижные оси декартовой системы координат. Расчет проводился при  $x_0 = -0.5 \text{ м}$ ,  $y_0 = -1 \text{ м}$ ,  $x_\tau = 0.34 \text{ м}$ ,  $y_\tau = 0.9 \text{ м}$ ,

$$\tau = 4 \text{ с}, k_{1(2,3)} = 1.$$

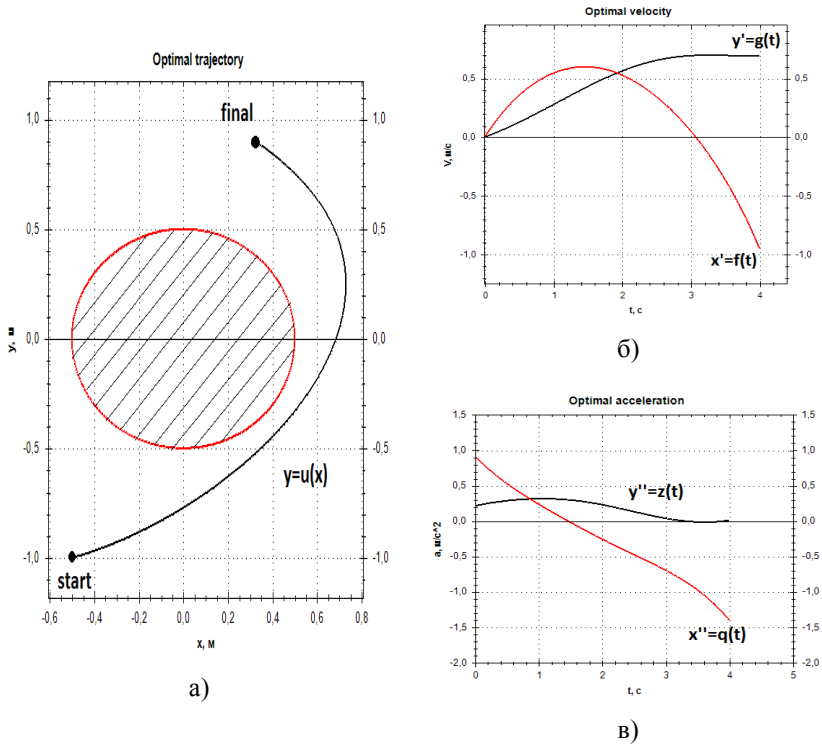


Рис. 1. Имитация работы системы планирования движения БТС

## 2. Некоторые аспекты практической реализации метода

Практическая отработка метода планируется на лабораторной модели БТС (Рис. 2.), которая представляет собой колесную платформу без поддресоривания с системой поворота задних колес сходной автотранспортной (наличие рулевой трапеции) и передним приводом. При этом на платформе установлена сенсорная группа, состоящая из трех лазерных дальномеров, для детектирования расстояний до поверхности препятствия. Как устройство поворота колес, так и тяговый электропривод оснащены угловыми энкодерами. Все сенсоры и датчики интегрированы с бортовой частью системы управления, реализованной на Raspberry Pi. Обмен информацией между бортовой и удаленной частями системы управления осуществляется через Wi-Fi-канал связи.



Рис. 2. Лабораторная модель БТС

Лабораторная модель обладает протяженностью в отличие от материальной точки, характеризующейся в пространстве числовыми координатами. Но представление БТС в виде материальной точки и решение задачи отсутствия столкновений корпуса робота с поверхностью препятствия совместно возможны. Для этого в центре масс робота строится окружность радиуса  $R_1$ , полностью охватывающая корпус БТС без пересечения. При обнаружении препятствия в виде трех отраженных сигналов лазерной группы, трем точкам поверхности препятствия ставится в соответствие окружность радиуса  $R_2$ . После чего радиусы  $R_1$  и  $R_2$  суммируются  $R = R_1 + R_2$ , а их суммарное значение используется в формулах, содержащих переменную  $R$ . Т. е. границы окружности, моделирующей препятствие, искусственно расширяются, тем самым, при выполнении условия (6), гарантируя отсутствие столкновений. Данный подход может быть использован и для планирования движения шагающих машин [Леонард, 2013], [Briskin et. al., 2014b], [Briskin et. al., 2010], [Briskin et. al., 2013], [Briskin et. al., 2003]. Необходимость в рассмотрении БТС как твердого тела (системы твердых тел, обладающих массами и моментами инерции) возникает в том случае, когда конструкция накладывает ограничения на физическую реализацию траекторий движения, а количество неизвестных силовых факторов (управляющих сил и реакций) превышает значение два. Что имеет место быть для лабораторной модели. Задача определения управляющих сил (напряжений) необходима для экспериментальной отработки движения, но представляет собой вопрос, требующий отдельного детального рассмотрения. Современный уровень развития цифровой техники, позволяет систему управления (СУ) робота разделить на две составляющие: бортовую СУ [Павловский и др., 2016] и удаленную, обменивающие между собой управляющей информацией. При

этом комплекс расчетов, связанный с планированием движения самим роботом, можно реализовывать на удаленной ЭВМ со значительной оперативной памятью, а для бортовой СУ оставить обработку сформированных управляющих сигналов.

## Заключение

При построении критерия оптимальности, входящего в базу знаний робота [Леонард и др., 2015], [Леонард и др., 2017], [Брискин и др., 2016] в виде дифференциальных уравнений движения, для реального движения БТС потребуются учет конструктивных особенностей робота таких, как минимально допустимый радиус поворота или «разрешенное изменение в пространстве» вектора скорости центра масс БТС. Введение подобных частных показателей будет усложнять вид дифференциальных уравнений движения, поэтому необходимо исследовать вопрос корректной применимости численных методов при синтезе законов движения по заданным оператором граничным условиям. Следующим шагом по интеллектуализации БТС в рамках предлагаемого подхода по самопланированию движения может стать автоматизация получения дифференциальных законов движения самим БТС по заданному оператором критерию – целевой функции, которая своей структурой и набором параметров отражает специфику планируемого поведения робота. При этом специфика поведения робота, как было показано в статье, зависит от значений весовых коэффициентов, которые меняются самой машиной при изменении условий среды, а по необходимости и при синтезе законов движения.

Авторы статьи не претендуют на полноту предлагаемого метода. По мере получения новых теоретических выводов и экспериментальных результатов метод будет совершенствоваться и распространяться как на класс статических препятствий сложной конфигурации, так и на класс динамических препятствий. Что потребует использование более мощной сенсорной базы класса LIDAR.

В статье, для иллюстрации работы (Рис. 1) системы управления БТС, использовалась авторская программа, разработанная на языке Visual C++ в среде разработки Microsoft Visual Studio 2010. Предварительный расчет параметров осуществлялся в лицензионной программе MathCad Education (в рамках закупки ПО по программе подготовки кадров для ОПК инфраструктурного конкурса МИНОБРНАУКИ от 2014 г.).

## Список литературы

- [Брискин и др., 2011] Брискин Е.С., Калинин Я.В. Об энергетически эффективных алгоритмах движения шагающих машин с цикловыми движителями // Известия Российской академии наук. Теория и системы управления – 2011. – № 2.
- [Брискин и др., 2016] Walking robot «character» as element of intelligent system / Е.С. Брискин, А.В. Малолетов, Н.Г. Шаронов, Я.В. Калинин, А.В. Леонард, В.А. Серов, В.А. Шурыгин // Advances in Cooperative Robotics : Proceedings of the International Conference on CLAWAR 2016 (London, UK, 12-14 September 2016) / ed. by Mohammad O. Tokhi, Gurvinder S. Virk. – [Published by World Scientific Publishing Co. ; printed in Singapore, 2016]. – P. 386-394.
- [Васильев, 2012] Васильев, Е.М. Система оптимального управления подвешенным грузом // Вестник ВГТУ. – 2012. – № 1.
- [Жога и др., 2010] Жога, В.В., Гаврилов А.Е., Еременко А.В. Оптимальный закон горизонтального перемещения мобильного робота с ортогональными шагающими движителями // Известия ВОЛГТУ. – 2010. – № 8.
- [Казаков и др., 2016] Казаков К.А., Семенов В.А. Обзор современных методов планирования движения // Труды Института системного программирования РАН. 2016. Т. 28. № 4. С. 241-294.
- [Лавренов и др., 2016] Лавренов Р.О., Афанасьев И.М., Магид Е.А. Планирование маршрута для беспилотного наземного робота с учетом множества критериев оптимизации // Третий Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2016, 22-23 сентября 2016 г., г. Иннополис, Республика Татарстан, Россия): Труды семинара. – М: Изд-во «Перо», 2016. – 184 с. С. 10-20.
- [Леонард, 2013] Леонард, А.В. Цикловый шагающий движитель с направляющими. Свойства. Управление. Пути совершенствования // Известия ВолгГТУ. Серия "Актуальные проблемы управления, вычислительной 3 техники и информатики в технических системах". Вып. 16 : межвуз. сб. науч. ст. / ВолгГТУ. - Волгоград, – 2013. – № 8 (111).
- [Леонард и др., 2015] Леонард А.В., Горбов И.А., Калинин Я.В., Брискин Е.С. Интеллектуализация системы управления робота, моделируемого материальной точкой на плоскости // Известия Волгоградского государственного технического университета. 2015. № 14 (178).
- [Леонард и др., 2017] О влиянии «характера» группы роботов на их движение / А.В. Леонард, Е.С. Брискин, И.А. Горбов // Известия ВолгГТУ. Сер. Актуальные проблемы управления, вычислительной техники и информатики в технических системах. - Волгоград, 2017. - № 1 (196). - С. 116-120.
- [Ловейкин и др., 2014] Ловейкин В.С., Шевчук О.Г. Влияние параметров крановой тележки с гибким подвесом груза на оптимальный динамический режим движения // Вестник ХНАДУ. – 2014. – № 65-66.
- [Мартыненко, 2005] Мартыненко Ю.Г. Управление движением мобильных колёсных роботов // Фундаментальная и прикладная математика. 2005. Т. 11. № 8. С. 29-80.
- [Михайлов и др., 2016] Михайлов Б.Б., Назарова А.В., Ющенко А.С. Автономные мобильные роботы - навигация и управление // Известия ЮФУ. Технические науки. 2016. № 2 (175). С. 48-67.
- [Охоцимский и др., 1957] Охоцимский, Д.Е., Энеев Т.М. Некоторые вариационные задачи, связанные с запуском искусственного спутника Земли // Успехи физических наук. – 1957.



- [Павловский и др., 2016] Павловский В.Е., Кий К.И., Орлов И.А., Алисейчик А.П. Информационная система интеллектуального беспилотного автомобиля "АВТОНИВА" // Третий Всероссийский научно-практический семинар «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2016, 22-23 сентября 2016 г., г. Иннополис, Республика Татарстан, Россия): Труды семинара. – М: Изд-во «Перо», 2016. – 184 с. С. 88-97.
- [Юревич и др., 2007] Юревич Е.И. Интеллектуальные роботы // М.: Машиностроение, 2007.
- [Briskin et. al., 2003] Briskin E.S., Chernyshev V.V., Maloletov A.V. Power efficiency and control algorithms of walking machine with cycle propellers // CLAWAR 2003. – 2003.
- [Briskin et. al., 2014a] Briskin E.S., Kalinin Ya. V., Maloletov A. V., Chernyshev V.V. On the energy efficiency of cyclic mechanisms // Mechanics of Solids. – 2014. – Vol. 49, No. 1.
- [Briskin et. al., 2013] Briskin E.S., Leonard A.V. Stability of translational motion of a walking machine with cyclic drives // Journal of Computer and Systems Sciences International. – 2013. – Vol. 52, No. 6.
- [Briskin et. al., 2014b] Briskin E.S, Shurygin V.A., Chernyshev V.V., Maloletov A.V., Sharonov N.G., Kalinin Ya. V., Leonard A.V., Serov V.A., Mironenko K.B., Ustinov S.A. Problems of Increasing Efficiency and Experience of Walking Machines Elaborating // ROMANSY 2014.
- [Briskin et. al., 2010] Briskin E.S., Zhoga V.V., Chernyshev V.V., Maloletov A.V., Kalinin Ya.V., Sharonov N.G. Walking machines (elements of theory, experience of elaboration, application) // CLAWAR 2010. – 2010.
- [Darintsev et. al., 2007] Darintsev O. V. and Migranov A. B. Genetic Algorithms-Based Path-Planning System for the Motion of a Group of Mobile Micro-robots // Journal of Computer and Systems Sciences International. – 2007. – Vol. 46, No. 3.
- [Khatib, 1986] Khatib O. Real time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics and Research, vol. 5, № 1, 1986, pp. 90–98.
- [Koyasu et. al., 2006] Hiroshi Koyasu and Jun Miura Mobile robot motion planning considering path ambiguity of moving obstacles // Intelligent Autonomous Systems. – 2006. – No. 9.
- [Rui et. al., 2009] Lin Rui, Du Zhijiang and Sun Lining Moving Object Tracking based on Mobile Robot Vision" // ICMA. – 2009.
- [Statnikov et. al., 1995] Statnikov R.B. Matusov J.B. Multicriteria Optimization and Engineering. – Springer, 1995
- [Uchibe et. al., 2000] Eiji Uchibe, Masakazu Yanase, and Minoru Asada Behavior generation for a mobile robot based on the adaptive fitness function // IAS-6. – 2000.

УДК 004.896

## ФОРМИРОВАНИЕ СЕМАНТИЧЕСКИХ ОПИСАНИЙ ДЛЯ РЕШЕНИЯ ЦЕЛЕВЫХ ЗАДАЧ АВТОНОМНЫМИ БЕСПИЛОТНЫМИ ЛЕТАТЕЛЬНЫМИ АППАРАТАМИ

Н.В. Ким (*nkim2011@list.ru*)Н.Е. Бодунков (*boduncov63@yandex.ru*)

Московский авиационный институт, Москва

**Аннотация.** Расширение круга решаемых задач беспилотными летательными аппаратами (БЛА) требует совершенствования бортового алгоритмического обеспечения. Данная работа посвящена проблемам формирования семантических описаний наблюдаемой сцены, предназначенных для повышения эффективности решения целевых задач автономными БЛА в помещениях. Например, при почтовой доставке малогабаритных грузов (с помощью БЛА) в качестве конечной точки может быть указан «стол» или другие объекты, требующие их семантического описания. Предлагаемое семантическое описание окружающей среды основано на распознавании объектов, присутствующих на наблюдаемой сцене, с помощью технологий технического зрения. При этом из исходного растрового 3D изображения, получаемого от лазерного сканера или стереосистемы, на первом этапе формируется векторное описание, а затем искомое семантическое описание. Представлена методика формирования семантических описаний на основе использования 3-х мерных векторных описаний, подготовленных онтологических описаний объектов сцены и продукционных правил.

**Ключевые слова:** беспилотный летательный аппарат, анализ изображения, семантическое описание наблюдаемой сцены.

### Введение

В настоящее время актуальным и практически важным является использование автономных роботов, в частности, беспилотных летательных аппаратов (БЛА), в различных задачах наблюдения и доставки грузов [Feng, 2009]. При этом автономные БЛА, должны реализовывать собственное управление и решать поставленные задачи без непосредственного участия оператора.

Отдельным направлением применения автономных роботов является решение целевых задач в ограниченных пространствах и помещениях.

Целевыми задачами в этих случаях являются, например, задачи, связанные с помощью при ликвидации последствий чрезвычайных происшествий (пожаров, аварий, взрывов), доставка полезных грузов, решение антитеррористических задач и пр.

Эффективными методами поиска искомых объектов являются подходы, основанные на предварительном определении локальных областей интереса [Kim, 2015b] с использованием семантической информации о состоянии текущей наблюдаемой сцены. Семантическое описание объектов производится с использованием технологий компьютерного зрения, теории распознавания [Kim, 2014].

Базовым видом описаний наблюдаемой сцены являются растровые изображения, содержащие информацию от бортовых систем наблюдения, оснащенных датчиками различной физической природы [Zhang, 2012]. На основе растровых данных можно получить более компактные векторные описания [Загоруйко, 2014]. При решении ряда целевых задач использование растровых и векторных описаний недостаточно. В этих случаях возникает необходимость их дополнения и расширения семантическими описаниями [Поспелов, 1986].

## 1. Формирование семантических описаний

Основой для получения семантического описания является растровое изображение, получаемое, например, с помощью: дальномертрических систем, стереозрения или инфракрасных 3D-сканеров.

На рис.1 представлена исходная наблюдаемая сцена.



Рис. 1. Наблюдаемая сцена

Перевод дальнометрического изображения (исходного облака точек), полученного, например, с помощью системы Kinect, в векторное описание производится на основе известных уравнений, описывающих плоскости [Загоруйко, 2014]. В результате анализа координат точек определяются коэффициенты уравнений отдельных плоскостей и их границы.

На рисунке 2 представлено изображение с выделенными плоскостями.

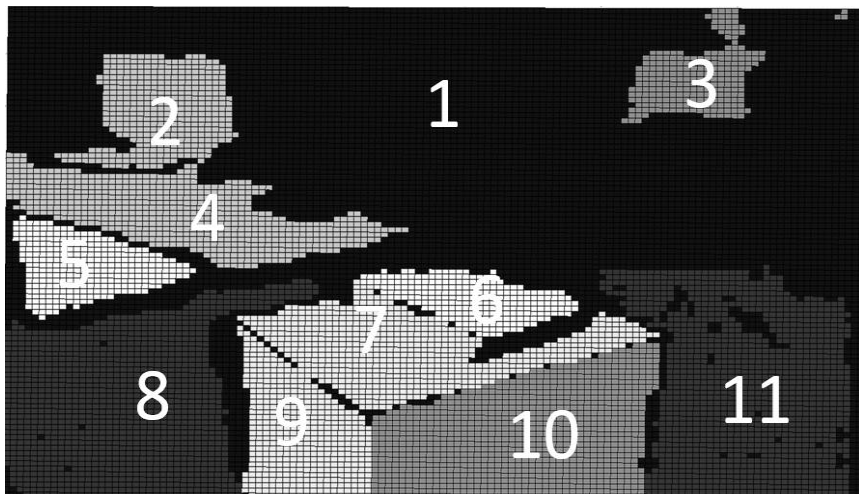


Рис. 2. Плоскости полученные, в результате кластеризации

На наблюдаемой сцене были выделены 11 плоскостей, описываемых уравнениями с уникальными коэффициентами. В рассматриваемом случае набор полученных коэффициентов является векторным описанием наблюдаемой сцены. В качестве примера ниже представлен фрагмент данного описания:

- плоскость 4:  $A_4=-16.26, B_4=-295.9, C_4=33.45, D_4=255.6$ ;

.....  
- плоскость 7:  $A_7=-4.21, B_7=-216.67, C_7=27.6, D_7=71.3$  ;

.....  
- плоскость 9:  $A_9=-90.3, B_9=17.59, C_9=16.47, D_9=75.4$ .

Семантические описания используют векторные и растровые описания объектов, но наделяют их дополнительными («смысловыми») атрибутами и отношениями [Поспелов, 1986]. Семантические описания являются ключевыми элементами анализа ситуации, который необходим для решения ряда целевых задач автономными БЛА [Kim, 2015a].

Семантические описания носят иерархический характер и в зависимости от уровня иерархии делятся на классы, подклассы и разделы различной степени детализации [Oberle, 2009].

Рассмотрим процедуру формирования семантического описания фрагмента помещения, соответствующего рис. 1.

Предполагается, что:

исходный алфавит классов объектов заранее определен;

словарь признаков (атрибутов) сформирован;

векторное описание плоскостей составлено и коэффициенты соответствующих уравнений известны.

Методика формирования описаний включает следующие этапы:

Определяются положения выделенных плоскостей;

Производится возможная принадлежность выделенных плоскостей соответствующим классам объектов;

Определяются отношений между объектами.

В общем случае могут использоваться различные виды отношений ситуации: пространственные, пространственно-временные, временные, каузальные. Если предполагаемое решение поставленных задач основано на анализе расположения объектов интереса, т.е. пространственных отношений между объектами, то целесообразно использовать пространственные описания.

Пусть известно, что в исследуемом пространстве расположены объекты: Стена, Пол, Стол, Тумба, Монитор, книги. Примеры их эталонных семантических описаний представлены в таблице 1.

Таблица 1

Семантическое описание искомых объектов

№	Класс объекта	Атрибут (размеры в мм)	Отношения атрибутов	Межобъектные отношения
1	Стена	Плоскость Ст, размеры не определены		Расположение вертикальное
2	Пол	Плоскость П, размеры не определены		Расположение горизонтальное, параллельно и ниже на 1600 мм плоскости X0Z
3	Стол	Плоскость С1: 600x1200;	С1 перпендикулярно С2, С3, С4;	С1 параллельно полу и выше на 750 мм

		Плоскости С2, С3: 600x750; Плоскость С4: 800x400;	С2 параллельно С3; С4 перпендикулярно С2, С3	
4	Тумба	Плоскость Т1: 550x400; Плоскости Т2, Т3: 600x400; Плоскости Т4, Т5: 600x550;	Т1 перпендикулярно Т2, Т3, Т4, Т5; Т2 параллельно Т3; Т4 параллельно Т5;	Т1 параллельно полу и выше на 600 мм
5	Монитор	Плоскость М1: 400x350;		Расположение М1 – на С1, перпендикулярно С1
6	Книги	Плоскость К1: от 150x200 до 250x300		Расположение К1 – на С1 или Т1; параллельно С1 или Т1

Распознавание объектов может производиться на основании заранее сформированных правил [Kim, 2015b] и онтологий.

Например,

**Правило i:**

**If**

Определена плоскость **and** плоскость расположена горизонтально **and** плоскость расположена выше плоскости пола на 750 мм **and** размеры плоскости 600x1200 мм

**Then**

Плоскость относится к классу «Стол»;

**Правило i+1:**

**If**

Определена плоскость **and** класс плоскости не определен **and** плоскость расположена горизонтально **and** плоскость расположена выше плоскости пола на 600 мм **and** размеры плоскости 550x400 мм

**Then**

Плоскость относится к классу «Тумба»;

**Правило i+2:**

**If**

Определена плоскость **and** плоскость расположена вертикально в класс плоскости не определен **and** плоскость расположена выше плоскости пола на 600 мм от 750 мм **and** размеры плоскости 400x350 мм

### **Then**

Плоскость относится к классу «Монитор».

После распознавания объектов сцены определяются отношения между объектами и их свойства.

В результате формируется семантическое описание.

Пример семантического описания сцены (рис. 1, 2):

1. Распознанные объекты:

Стол, стена, монитор 1, монитор 2, тумба, книги.

2. Расположение объектов (относительное):

- стол - пол;
- стол - стена;
- монитор 1 - стол;
- тумба - пол;
- тумба - стол;
- книга - тумба.

## **Заключение**

Показано, что описания наблюдаемых сцен могут быть представлены в семантическом виде, позволяющем повысить эффективность решения ряда целевых задач автономными БЛА.

Предлагается методика формирования семантических описаний на основе использования 3-х мерных векторных описаний, подготовленных заранее онтологических описаний объектов сцены и производственных правил.

Рассмотрены процедуры получения 3-х мерных семантических описаний, предназначенных для организации полетов и решения целевых задач БЛА в помещениях.

## **Список литературы**

- [Feng, 2009] Lin Feng, Lum Kai-Yew, Chen Ben M., Lee Tong H. "Development of a vision-based ground target detection and tracking system for a small unmanned helicopter." Science in China Series F: Information Sciences, 2009, Springer.
- [Kim, 2015a] Kim N., Chervonenkis M. "Situational control unmanned aerial vehicles for traffic monitoring." Modern Applied Science, Vol. 9, No. 5, May 2015, Special Issue//Canadian Center of Science and Education. ISSN (printed): 1913-1844. ISSN (electronic): 1913-1852

- [**Türmer, 2011**] Türmer, S.; Leitloff, J.; Reinartz, P.; Stilla, U. (2011): Evaluation of selected features for car detection in aerial images. ISPRS Hannover Workshop 2011, 14.-17. Jun. 2011, Hannover.
- [**Zhang, 2012**] J. Zhang, L. Liu, B. Wang, X. Chen, Q. Wang, and T. Zheng, "High speed automatic power line detection and tracking for a UAV-based inspection," in International Conference on Industrial Control and Electronics Engineering (ICICEE), 2012, pp. 266-269
- [**Kim, 2014**] Kim N., Bodunkov N. «Computer Vision in Advanced Control Systems: Innovations in Practice», Volume 2, Editors M. Favorskaya, Lakhmi C. Jain, Springer 2014. – 295 p.
- [**Загоруйко, 2014**] Загоруйко С.Н., Казьмин В.Н., Носков В.П. Навигация БПЛА и 3D реконструкция внешней среды по данным бортовой СТЗ. // М.: «Новые технологии», ж. «Механика, автоматизация, управление» №8, 2014. С. 62-68.
- [**Kim, 2015b**] Kim N. Automated Decision Making in Road Traffic Monitoring by on-Board Unmanned Aerial Vehicle System. Indian Journal of Science and Technology, Vol 8(S10), December 2015.
- [**Горелик, 2004**] Горелик А.Л., Скрипкин В.А. Методы распознавания. М.: Высшая школа, 2004.
- [**Поспелов, 1986**] Поспелов Д.А. Ситуационное управление: теория и практика.- М.: Наука, - Гл.ред.физ.-мат.лит. 1986.-288с.
- [**Liang, 2012**] Liang Li, Shuqiang Jiang, Qingming Huang. "Learning Hierarchical Semantic Description Via Mixed-Norm Regularization for Image Understanding." Multimedia, IEEE Transactions ...> Volume:14, Issue:5, 2012, p. 1401 – 1413.
- [**Oberle, 2009**] Oberle, D., Guarino, N., & Staab, S. (2009) What is an ontology?. In: "Handbook on Ontologies". Springer, 2nd edition, 2009.



УДК 007.5, 004.8

## ПОИСК СОГЛАСОВАННЫХ РЕШЕНИЙ ЧЕЛОВЕКОМ И РОБОТОМ. ПРОБЛЕМА МАНИПУЛИРОВАНИЯ

В.Н. Жидков (*vladimir\_zhidkov@mail.ru*)  
Московский авиационный институт, Москва

Н.В. Ким (*nkim2011@list.ru*)  
Московский авиационный институт, Москва

**Аннотация.** В статье рассматриваются вопросы общения человека и робота с целью выработки согласованных решений. Определена структура обработки информации робота, обеспечивающая эффективное принятие согласованных решений. Показано, что при решении различных целевых задач человеком и роботом, робот может реализовать манипулирование человеком.

**Ключевые слова:** человек, психоэмоциональное состояние, психологическое воздействие, робот, целевые задачи, принятие решение, правила манипулирования.

### Введение

В рамках данной статьи будем считать, что манипулированием является достижение своих собственных целей *манипулятором* за счет психологического воздействия на конкретного человека.

В будущем дальнейшее развитие робототехники приведет к широкому распространению различных роботов (Р): водителей, продавцов, строителей и пр., к которым человек (Ч) будет обращаться за помощью. При этом в процессе общений Ч и Р возможно возникновение коллизий, связанных с недостаточным пониманием Ч и Р, различием решаемых целевых задач и т.д. Если основные психоэмоциональные особенности людей в ближайшие десятилетия вряд ли будут существенно меняться, то роботы могут регулярно оснащаться все более совершенными программно-аппаратными средствами, обеспечивающими более эффективное выполнение требуемых правил поведения.

С учетом данного фактора актуальным представляется вопрос о том, как должен быть оснащен Р, чтобы наиболее эффективно находить совместно с Ч согласованные решения. Если человек при выполнении каких-либо задач

будет обращаться за помощью к роботам, имеющим другие целевые задачи, то и человек, и робот должны будут принимать некоторые, устраивающие обоих, согласованные решения. При этом предполагается, что каждый из них будет стремиться в наибольшей степени обеспечить выполнение собственной целевой задачи.

Термин «согласованность» подразумевает, что Ч в процессе общения может пойти на некоторые уступки, т.е. изменить свои исходные цели с учетом имеющихся целевых функций Р. Таким образом, Р, целенаправленно изменяя исходные цели Ч с учетом своих интересов, является *манипулятором*.

## 1. Концепция построения комплексного алгоритма

Рассмотрим структуру комплексного алгоритма «**Принятие согласованных решений**», позволяющего Р «договариваться» с Ч о совместных действиях, а также манипулировать человеком.

В «базовом» случае предполагается, что цели Ч и Р совпадают. Тогда в Р может быть реализован вариант, в котором после получения запроса ( $Q$ ) от Ч формируется набор возможных совместных действий  $U(Q)$ . Если Ч выбирает некоторый  $u_i \in U(Q)$ , то считается, что согласованное решение принято.

Если цели Ч и Р не совпадают, то необходимо находить компромиссное решение, например, с использованием аппарата теории игр.

Одним из возможных вариантов нахождения согласованного решения является использование Р методов манипулирования. Наиболее просто реализуется вариант аналогичный классическому «программному» управлению. Здесь робот предварительно оснащается *набором приемов психологического воздействия (манипулирования) (ППВ)*, которые могут им использоваться. Существует большое количество подобных приемов [1] и их программно-алгоритмическая реализация вполне возможна.

Рассмотрим ситуацию, когда робот-водитель такси с целью повышения прибыли пытается уговорить пассажира выбрать более длинный маршрут (к заказанному пункту назначения), чем маршрут  $M(H)$ , выбранный «Навигатором». В этом случае может, например, использоваться прием «*Навязывание выбора*».

Алгоритм реализации приема состоит в следующем:

Из имеющегося набора возможных маршрутов выбираются маршруты  $M(1)$  и  $M(2)$  более длинные, чем  $M(H)$ , и проходящие через какие-либо достопримечательности. Пассажиру предлагается: «Если хотите увидеть уникальное здание и площадь, то поедem маршрутом  $M(1)$ , а если: парк и озеро, то  $M(2)$ ». На подобное предложение некоторые из пассажиров

выберут один из предлагаемых вариантов, что позволяет достичь  $P$  своей цели.

Эффективность воздействия приемов манипулирования на различных людей различна. В существенной степени она зависит от психотипа конкретного человека и его эмоционального состояния [Бодалев, 1995].

Поэтому в следующем варианте более эффективного манипулирования учитывается психотип человека.

Известны методы определения психотипов по внешним признакам. Например, в [3] выделяются признаки: "сутулый, выпяченный живот, худой, интуиция +1, логика +1, экстраверсия +1" или "неулыбчив, сжатые губы, сухое выражение лица - логика +1", по которым делается заключение о психотипе данного Ч. Подобные признаки могут быть выделены с использованием известных технологий компьютерного зрения на базе различных методов распознавания [Kim N и др., 2014].

Данные о психотипе человека должны собираться и храниться в психологическом паспорте человека (ППЧ( $n$ )), где  $n = 1, \dots, N$  индекс конкретного человека. Эти данные позволят формировать соответствующие индивидуальные ППВ( $n$ ).

Известно, что решения, который принимает Ч в существенной степени зависят от его эмоционального состояния [[Бодалев, 1995], поэтому логично в процессе манипулирования учитывать и этот фактор.

Для оценки состояния человека робот должен быть *оснащен модулем оценки эмоционального состояния Ч*.

В настоящее время уже разработаны программы, которые позволяют автоматически оценивать эмоциональное состояния людей на основе анализа изображений их лиц. Например, система видео-аналитики «Монитор Эмоций» [5] (бесплатная версия «ЭмоСкоп») позволяет в режиме реального времени контролировать **эмоциональный фон** и другие характеристики человека по выражению лица с использованием экспертной системы.

Учет эмоционального состояния позволит использовать более эффективные ППВ( $n, e, t$ ), где  $e$  – индекс эмоционального состояния,  $t$  – текущий момент времени.

Таким образом, в ППЧ необходимо вносить сведения о психотипе и эмоциональном состоянии Ч (в соответствии с его реакцией на определенные раздражители).

## 2. Модель взаимодействия Ч и Р

Рассматриваемая модель взаимодействия робота и человека разработана в среде MATLAB с использованием подсистемы моделирования динамических процессов SIMULINK. Модель разработана

в соответствии с принципами построения адаптивных психофизических интерфейсов, способных подстраиваться под конкретного человека-оператора и учитывающих в процессе общения его эмоциональное состояние [Жидков и др., 1999], [Жидков, 2013].

Процесс общения человека и робота рассматривается как непрерывный процесс устранения рассогласования между текущим условным состоянием некоторой потребности человека, и желаемым (мотивированным) состоянием этой потребности.

В модели условное состояние потребности человека кодируется некоторым числовым значением, абсолютная величина которого не несет никакой смысловой нагрузки - важна степень рассогласования между текущим и целевым значениями этой потребности (рассогласование целевой функции). Целевое значение потребности (целевая функция) определяется мотивацией человека, но робот может пытаться изменить ее по своему усмотрению (см. выше).

Модель реализована на основе нейро-нечетких систем ANFIS [Жидков, 2013], удобный инструмент для проектирования которых предлагается в пакете расширения MATLAB - Fuzzy Logic Toolbox.

Нейро-нечеткая система имеет базу знаний, содержащую нечеткие правила, формируемые экспертом, в которых заключен основной интеллект системы.

Кроме того, нейро-нечеткая система ANFIS способна обучаться на примерах решения задачи, которые могут быть получены от эксперта либо самостоятельно роботом в процессе адаптации его к окружающему миру.

Разработанная модель взаимодействия робота и человека содержит три контура:

*1. Контур динамики общения.* Содержит модуль человека (рис.1) и модуль "прикладного" интеллекта, который необходим роботу для решения целевой задачи (рис.2). Модуль человека содержит две нейро-нечеткие системы - ANFIS 1 и ANFIS 2. В базах знаний этих систем в виде нечетких продукций заложена информация о психотипе конкретного человека, а также информация, необходимая для моделирования эмоционального состояния человека в процессе общения с роботом.

Система ANFIS 1 имеет четыре входа, на основе которых определяется выходной параметр системы - целевая функция человека:

1. Целевой стимул (мотивация);
2. Мешающие факторы (шум);
3. Сигнал с выхода Out 1 модуля управления поведением человека (см. ниже);
4. Время общения с роботом.

Система ANFIS 2 предназначена для моделирования психотипа человека и фактически регулирует коэффициент демпфирования в контуре динамики общения. Система также имеет четыре входа:

1. Начальное демпфирование;
2. Время общения с роботом;
3. Информация от робота (In1), на который поступает сигнал с выхода Out 1 модуля управления эмоциями человека (см. ниже);
4. Текущее рассогласование целевой функции.

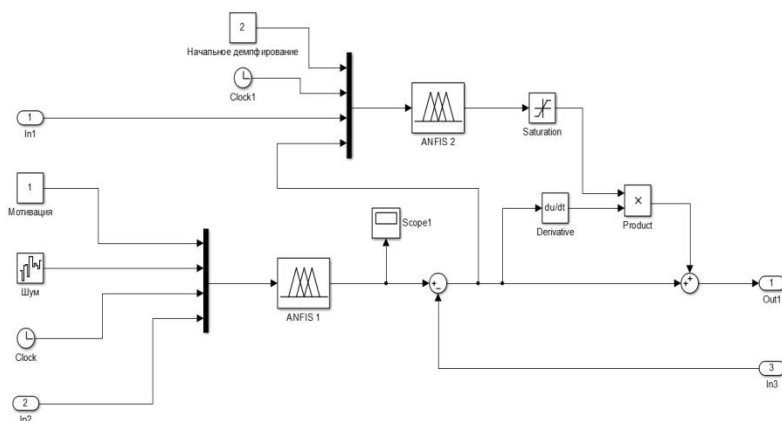


Рис. 1. Модель человека

Модуль прикладного интеллекта робота (рис.2) содержит нейро-нечеткую систему ANFIS3, в которой в виде нечетких продукций заложены знания, необходимые для устранения рассогласования между текущим и целевым состоянием человека и/или робота. Сигнал In1 модуля поступает сигнал с выхода Out1 модели человека (демпфированное рассогласование целевой функции). Этот сигнал дважды интегрируется (идет накопление информации) и затем поступает на "лингвистический" регулятор, реализованный на нейро-нечеткой системе, которая формирует информационный сигнал человеку (выход Out1), необходимый для удовлетворения текущей условной потребности человека, - сигнал обратной связи. С выхода Out1 сигнал обратной связи поступает на вход In3 модели человека - вход прямого восприятия информации от робота.

На вход блока ANFIS 3 поступают 3 сигнала:

1. Однократно проинтегрированный сигнал о рассогласовании целевой функции;

2. Двухкратно проинтегрированный сигнал о рассогласовании целевой функции;
3. Мотивация робота, которая задается внешней системой управления роботом и может отличаться от мотивации человека. Сигнал мотивации робота с выхода Out2 также подается в модуль управления мотивацией человека ( см. ниже).

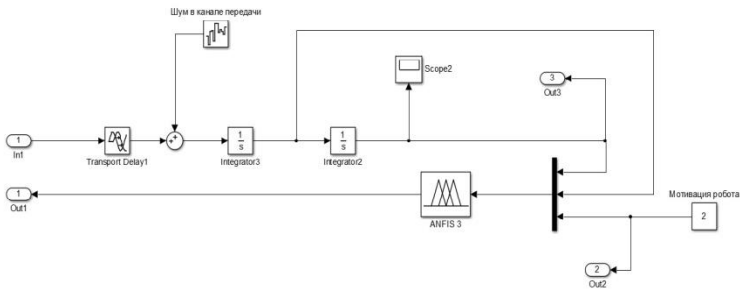


Рис. 2 Модель прикладного интеллекта

*2.Контур управления эмоциональным состоянием человека.* Содержит модуль определения диссонанса (рис.3) и модуль управления эмоциями человека (рис.4). Здесь принято, что *диссонанс* является обобщенной характеристикой текущего психоэмоционального состояния человека и определяется как функция текущего рассогласования целевой функции, тенденции ее изменения во времени, а также времени общения человека и робота.

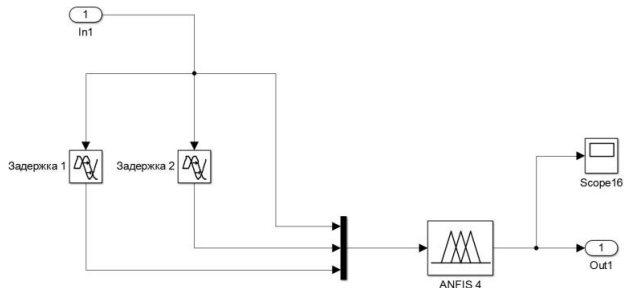


Рис. 3. Модель определения диссонанса

Нечеткие лингвистические правила определения диссонанса заложены в базу знаний нейро-нечеткой системы ANFIS 4.

На вход системы In1 с выхода Out 3 модуля прикладного интеллекта поступает двукратно проинтегрированный сигнал текущего рассогласования целевой функции;

Выход модуля Out1 непосредственно связан со входом In3 модуля управления эмоциями человека.

Модуль управления эмоциями человека (рис.4) реализован на нейро-нечеткой системе ANFIS5, которая содержит как базу знаний с нечеткими лингвистическими правилами коррекции психотипа человека, сформированными на основе общих знаний о методах управления эмоциями человека (ППВ), так и базу данных, содержащую индивидуальные психоэмоциональные паспорта конкретных людей (ППЧ( $n$ )). Выход системы Out1 поступает на вход In1 системы ANFIS 2, непосредственно моделирующей психотип человека.

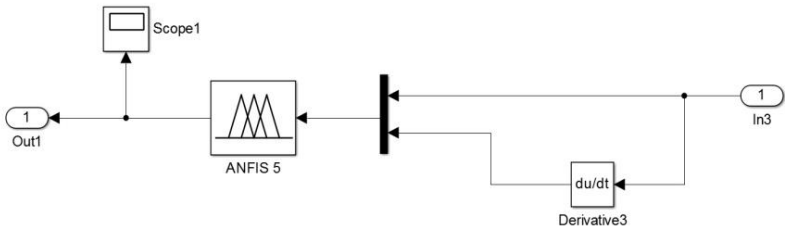


Рис. 4. Модель управления эмоциями человека

*3.Контур воздействия на мотивацию человека.* Модуль управления целевой функцией человека (рис.5) реализован на нейро-нечеткой системе ANFIS6, которая содержит как базу знаний с нечеткими лингвистическими правилами психологического воздействия (манипулирования) (ППВ) на человека с целью принудить его к изменению целевой функции (см. выше), так и базу данных, содержащую индивидуальные психоэмоциональные паспорта конкретных людей.

База знаний и база данных наполняется экспертными знаниями, и пополняется в процессе функционирования и самообучения робота. На вход системы In1 поступает информация о текущем диссонансе с выхода Out1 модуля определения диссонанса (рис.3) а также сигнал мотивации робота (вход In3) с выхода Out2 модуля прикладного интеллекта (рис.2).

Выход системы Out1 подается на вход восприятия информации от робота (In2) модуля человека (рис.1).

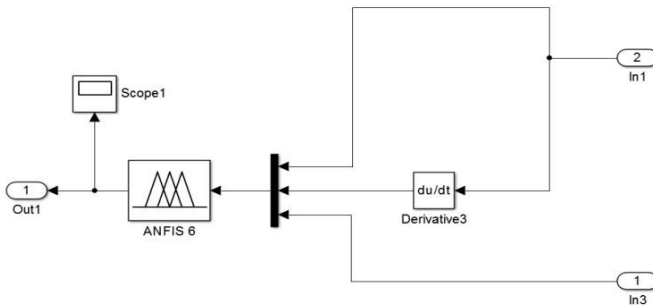


Рис. 5. Модель управления целевой функцией человека

### 3. Результаты моделирования

На рис. 6 и рис. 7 представлены результаты моделирования процесса общения человека и робота. В данном случае (ситуация 1) робот только помогает устранить рассогласование условной потребности человека (P). Исходное состояние  $P=0$ , желаемое состояние  $P=1$ . На рис.6 показан процесс согласования потребности, на рис. 7 - диссонанс.

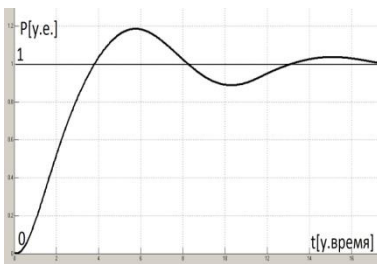


Рис. 6. Процесс согласования 1

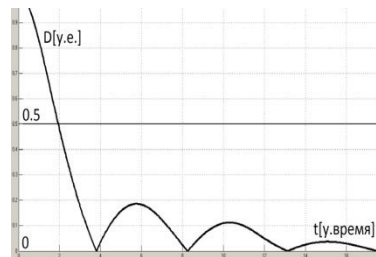


Рис. 7. Диссонанс согласования 1



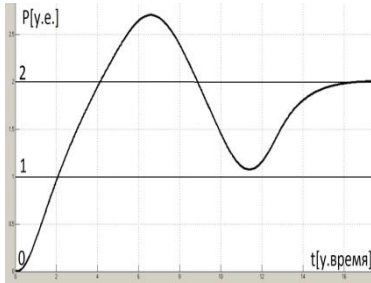


Рис. 8. Процесс согласования 2

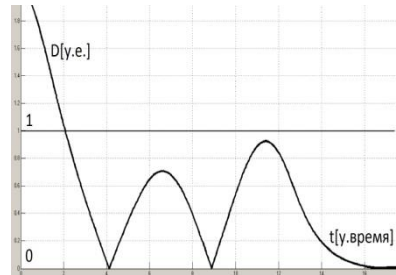


Рис. 9. Диссонанс согласования 2

Было проведено моделирование различных ситуаций, в частности, следующий сценарий общения человека и робота (рис. 6):

1. Робот предлагает вариант действий  $u_1$ ,
2. На  $t \geq 4$  человек принимает решение, что действие слишком рискованное,
3. На  $t \geq 6$  робот предлагает вариант  $u_2$ ,
4. На  $t \geq 8$  человек принимает решение, что действие недостаточно активно,
5. На  $t \geq 10$  робот предлагает вариант  $u_3$ ,
6. На  $t \geq 14$  человек принимает решение, что вариант удовлетворителен (рассогласование  $< 0,5$ ).

На рис. 8 и рис. 9 представлены результаты общения в ситуации (ситуация 2), когда робот навязывает свою мотивацию. Исходное состояние человека  $P=0$ , желаемое состояние человека  $P=1$ , мотивация робота  $P=2$ . Видно, что процесс идет с большими колебаниями диссонанса, человек сопротивляется мотивации робота, но в конце концов робот убеждает человека принять его (робота) мотивацию.

## Заключение

1. Показано, что возможно создание систем ИИ, которые смогут манипулировать человеком. При этом эффективность этих систем существенно возрастет, если на отдельных людей будут формироваться их индивидуальные психоэмоциональные паспорта (ППЧ);
2. Определена структура комплексного алгоритма «Принятие согласованных решений»;
3. Проведено моделирование процесса общения человека и робота, показавшее качественное сходство с реальным процессом;

4. Авторы считают, что использование подобных систем (особенно в сетевых технологиях) опасно с точки зрения возможности создания единых ППЧ и распространения технологий манипулирования людьми.

## Список литературы

- [Бодалев, 1995] Бодалев А. А. Личность и общение. — М.: Международная педагогическая академия, 1995. — 328 с. <https://bbf.ru/magazine/2/2746/>
- [Kim N и др., 2014] Kim N., Bodunkov N. «Computer Vision in Advanced Control Systems: Innovations in Practice», Volume 2, Editors M. Favorskaya, Lakhmi C. Jain, Springer 2014. – 295 p. <http://psychotype.ru/article/a-15.html>
- [Жидков, 2013] Жидков В.Н. Мягкие вычисления в интегрированных системах летательных аппаратов: Учеб.пособие.- М.: Изд-во МАИ, 2013.- 388с. <http://911.prolan.ru/applying/emotionsmonitor.html>
- [Жидков и др., 1999] Жидков В.Н., Жидков М.В, Красильщиков М.Н. Обучаемый интеллектуальный помощник пилота. Анализ вариантов реализации // Полет. 1999. №4

УДК 004.8, 681.5, 007.52

## ПОСТРОЕНИЕ МНОГОМЕРНОГО КЛАССИФИКАТОРА ИНТЕЛЛЕКТУАЛЬНЫХ РОБОТОВ

Г.В. Ройзензон (*rgv@isa.ru*)  
ИСА ФИЦ ИУ РАН, Москва  
МФТИ, Москва

В.Э. Карпов (*karpov\_ve@mail.ru*)  
НИЦ Курчатовский институт, Москва  
МФТИ, Москва

В.Е. Павловский (*vlpavl@mail.ru*)  
ИПМ им. М.В. Келдыша РАН, Москва  
МФТИ, Москва

В.Б. Бритков (*britkov@isa.ru*)  
ИСА ФИЦ ИУ РАН, Москва  
МФТИ, Москва

**Аннотация.** В работе предложен механизм построения многомерного классификатора интеллектуальных роботов (ИР), основанного на многокритериальном подходе. Критически проанализированы существующие варианты определения ИР. Предложено новое многокритериальное определение ИР. Проанализирован механизм, позволяющий усовершенствовать ИР, основанный на исследовании его свойств (оценок по критериям). Исследована проблематика анализа больших данных в интеллектуальной робототехнике. Рассмотрены способы верификации различных групп признаков, которые могут быть использованы для описания ИР. Представлен пример применения предлагаемого подхода.<sup>1</sup>

**Ключевые слова:** интеллектуальный робот, многомерный классификатор, большие данные, многокритериальный подход.

---

<sup>1</sup> Работа выполнена при финансовой поддержке РФФИ (проекты № 16-07-00865, 16-29-12901, 16-29-12878).

## Введение

К концу следующего десятилетия можно ожидать лавинообразный рост (несколько десятков миллиардов единиц) числа самых различных интеллектуальных устройств (ИУ) разного масштаба (интеллектуальных роботов (ИР), умных машин, умных предприятий, умных городов и т.п.). Соответственно подобное развитие событий приведет к необходимости обработки огромных массивов информации, поступающих от ИУ. В этой связи исследователи столкнутся с проблемой обработки больших данных (big data) совершенно другого масштаба даже по сравнению с текущим моментом. Под большими данными понимаются такие данные, объем которых превосходит текущие возможности оперирования ими в обозримый период. Важным направлением исследований является использование больших данных в рамках интерактивных компьютерных систем. Соответственно можно предложить альтернативное определение больших данных, формулируемое следующим образом: под большими данными понимаются массивы разнородной (структурируемой, слабо структурируемой и неструктурируемой) информации, которые не могут быть непосредственно использованы в человеко-машинных процедурах многокритериального принятия решений [Бритков, Ройзензон, 2015].

Как правило, большие данные характеризуются пятью характеристиками (5V), а именно: (1) Volume (физический объём); (2) Velocity (необходимость высокоскоростной обработки); (3) Variety (многообразие, необходимость обработки различных типов структурированных и полуструктурированных данных); (4) Veracity (достоверность); (5) Validity (обоснованность).

Огромное число ИУ и гигантские объемы информации требуют разработки новых подходов. В частности, необходимо разработать новый многомерный классификатор ИУ. Разделение ИУ на несколько категорий по степени интеллектуальности позволит дифференцировать гигантские потоки информации, например, по приоритету их обработки. Т.е. отслеживание состояния (и функции влияния извне) устройств, в которых полноценно реализована возможность планирования и выполнения действий, направленных на решение задачи является, безусловно, более важным, чем отслеживание состояния устройств, которые являются квазиинтеллектуальными. В качестве примера можно привести аварию с автомобилем Tesla, который управлялся автопилотом [Boudette, 2017], что привело к гибели водителя. Система управления автомобиля Tesla просто не успела правильно проанализировать аварийную ситуацию, что привело к трагическим последствиям. Можно только предполагать, каким образом нужно будет обрабатывать большие данные, когда на дорогах к концу следующего десятилетия появятся миллионы таких "умных" автомобилей

[Павловский и др., 2016] (разумеется, при сохранении современных тенденций и принципов обработки данных). Соответственно рост числа ИУ требует разработки совершенно новых подходов к оценке технологических рисков, в которых будут гармонично сочетаться возможности интеллектуальных устройств самостоятельно предотвращать какие-либо нежелательные последствия (аварии) и возможности человека вмешиваться в такие процессы. Очевидно, что роль человеческого фактора при оценке технологических рисков (статистические и экспертные подходы) будет, очевидно, в перспективе сокращаться (человек просто физически не успеет среагировать на различные опасные ситуации, которые могут возникать при использовании ИУ). Подобное развитие ситуации потребует проектирования ситуационных центров нового поколения [Бритков, Ройзензон, Фридман, 2016] для отслеживания состояния огромного числа интеллектуальных устройств для возможности оперативного вмешательства в их работу.

Как уже было отмечено, тенденция развития современной интеллектуальной робототехники неразрывно связана с постоянным усложнением как непосредственной «железной» составляющей, так и интеллектуальной «начинки» – программного обеспечения. В этой связи необходимо рассматривать такие сложные технические решения, как ИР, по частям [Крон, 1972], [Ройзензон, 2005]. Такой подход предполагает описание ИР через набор признаков (свойств, критериев и т.п.).

## **1. Многокритериальный подход к определению интеллектуального робота**

Современное состояние (стремительное развитие) интеллектуальной робототехники требует применения новых подходов как к определению, что же такое «интеллектуальный робот», так и к вполне самостоятельной задаче, а именно классификации ИР. С одной стороны, существует множество вариантов определения ИР (равно как и самого понятия робот). С другой стороны, попытки дать непротиворечивое определение этим понятиям чаще всего приводят к тому, что многие авторы вынуждены либо предлагать самые общие, сугубо качественные, понятийные определения, либо вовсе уходят от дефиниций. Это хорошо иллюстрирует анализ целого ряда классических работ в области робототехники [Янг, 1979], [Шахинпур, 1990], [Накано, 1988], [Каляев, Лохин, Макаров, 2007]. Немногим лучше обстоит дело и с попытками дать образные определения, "от общей постановки задачи" ([Павловский, 2014], [Добрынин, Карпов, 2006]).

Возможной основой для формирования механизма определения ИР может служить разработка системы признаков, характеризующих такие сложные системы. При этом очевидно, что большая часть признаков,

описывающих ИР, является качественной. Таким образом, разработка многомерного классификатора ИР может быть отнесена к слабоструктурированной задаче принятия решений [Simon, Newell, 1958], что предполагает применение методов вербального анализа решений (ВАР) ([Асанов и др., 2001], [Ларичев, 2006]) для построения такого классификатора.

В работе [Карпов, Павловский, Ройзензон, 2016] представлен ряд различных определений ИР, которые можно отнести к таким категориям, как перечислительное, когнитивное, функциональное и поведенческое (бихевиористское). Вместе с тем было показано, что наиболее конструктивным может считаться т.н. многокритериальное определение, основанное на применении многомерного классификатора [Карпов, Карпова, 2002]. Рассмотрим подробнее использование многокритериального подхода как для формализации определения ИР, так и для построения многомерного классификатора.

В известной книге Э. В. Попова [Попов, Фирдман, 1976] предложена система критериев, описывающая ИР. В нашем случае для построения полной непротиворечивой многокритериальной классификации используется метод ВАР ЦИКЛ [Асанов и др., 2001]. В качестве "осей" такого многомерного классификатора возьмем, например, следующие:

К1. Характеристика внешнего мира, в котором способен работать робот. Под моделью внешнего мира робота мы будем понимать множество объектов (включая самого робота) и отношений, заданных на этом множестве, которые описывают пространство возможных состояний среды, окружающей робота.

К2. Степень восприятия роботом информации о внешнем мире.

К3. Степень гибкости решающей программы роботов.

К4. Степень автономности робота (составной критерий). Под автономностью мы будем понимать способность робота самостоятельно, без участия человека, решать поставленные перед ним задачи. Здесь следует выделить такие аспекты, как мобильность, энергетическая и функциональная автономность.

На рис.1 представлена подробная иерархическая система критериев и оценок интеллектуального робота.

В результате работы метода многокритериальной классификации вырабатываются решающие правила, которые позволяют отнести тот или иной ИР к одному из классов решений. Например: класс А – Интеллектуальный робот; класс В – Квазиинтеллектуальный робот и класс С – «Обычный» робот. Использование такого многомерного классификатора применительно к задачам интеллектуального анализа больших данных, очевидно, позволит получить определенный выигрыш за счет дифференцирования гигантских потоков информации по приоритету

их обработки. Другим важным аспектом многомерной классификации является возможность введения некоей метрики, определения степени близости того или иного ИУ к интересующей целевой точке, области, классу.

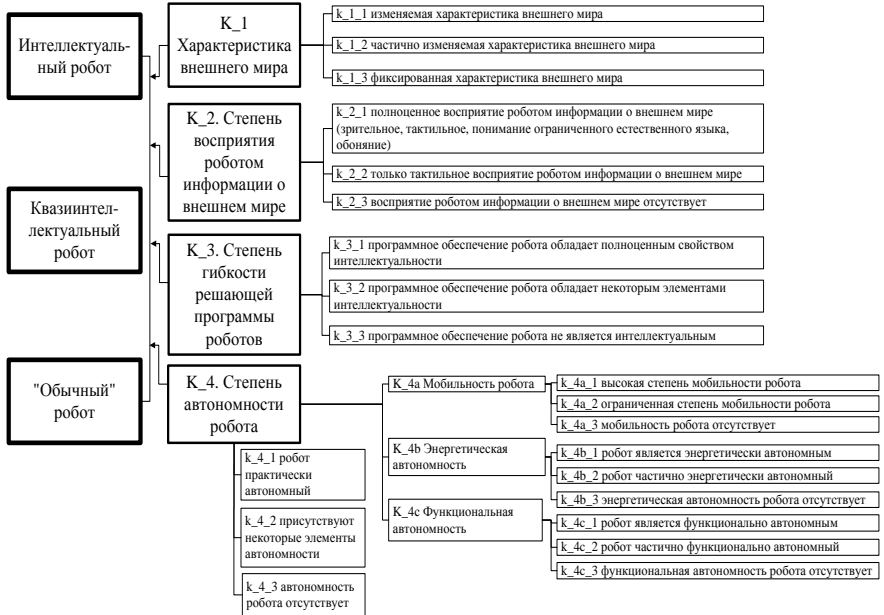


Рис. 1. Иерархическая система критериев интеллектуального робота

## Заключение

Неизбежный рост числа интеллектуальных устройств разного масштаба требует создания новых принципов их классификации. Разделение ИУ на несколько категорий по степени интеллектуальности позволит дифференцировать гигантские потоки информации, например, по приоритету их обработки. В работе предлагается один из возможных механизмов построения подобного многомерного классификатора, основанного на многокритериальном подходе.

Большинство имеющихся определений рассматривают ИР только с какой-то одной или нескольких позиций. В то же время активное развитие интеллектуальной робототехники требует рассмотрения функционала ИР с самых разных сторон, т.е. учета широко спектра признаков.

Использование метода ВАР многокритериальной порядковой классификации позволяет через свойства (признаки, критерии) ИР сформулировать прозрачное и понятное определение того, какой робот действительно может рассматриваться как интеллектуальный и почему, а какой является квазиинтеллектуальным или даже самым тривиальным («простым»). Такой подход позволяет при анализе оценок, составляющих границы классов решений, выявить, каким образом, улучшив те или иные, характеристики робота, его можно «передвинуть» в более высокую категорию (класс решений). Кроме того, при сравнении роботов одной и той же категории (относящихся к одному классу решений), такой подход позволяет определить, каким образом можно уменьшить стоимость аппаратно-программного решения робота без существенного ущерба для его интегральной функциональности.

Важной особенностью предложенного подхода является возможность сформировать разные наборы составных критериев различными способами (например, последовательно объединяя критерии попарно или формируя группы критериев исходя из некоторой смысловой общности). Это позволяет сравнить полученные результаты для разных вариантов классификации и выбора с целью оценки качества решения исходной проблемы, а также сравнить распределения ИР по классам решений для одного и того же набора составных критериев, сформированных с помощью различных подходов. Такая методология позволяет ЛПП выбрать как наиболее предпочтительный набор составных критериев, так и метод (совокупность методов) их построения в рамках решения конкретной практической задачи [Ройзензон, 2012]. Кроме того, подобный подход позволяет верифицировать те или иные группы признаков, которые могут быть использованы для описания ИР.



## Список литературы

- [Асанов и др., 2001] Асанов А.А. и др. Метод многокритериальной классификации ЦИКЛ и его применение для анализа кредитного риска // Экономика и математические методы. 2001. Т. 37. № 2. С. 14–21.
- [Бритков, Ройзензон, 2015] Бритков В.Б., Ройзензон Г.В. Анализ больших данных в наукометрии // Управление развитием крупномасштабных систем MLSD 2015. Материалы восьмой международной конференции в 2 т. / Под ред. С. Н. Васильева, А. Д. Цвиркуна, Т.2. М.: ИПУ РАН, 2015. С. 363–366.
- [Бритков, Ройзензон, Фридман, 2016] Бритков В.Б., Ройзензон Г.В., Фридман А.Я. Многокритериальный подход к оценке ситуационных центров // Проблемы прогнозирования чрезвычайных ситуаций. XV Всероссийская конференция. Сборник материалов. М.: ФКУ Центр «Антистихия» МЧС России, 2016. С. 26–28.
- [Добрынин, Карпов, 2006] Добрынин Д.А., Карпов В.Э. Моделирование некоторых форм адаптивного поведения интеллектуальных роботов // Информационные технологии и вычислительные системы. 2006. № 2. С. 45–56.
- [Каляев, Лохин, Макаров, 2007] Каляев И.А., Лохин В.М., Макаров И.М. Интеллектуальные роботы: учебное пособие для вузов / Под ред. Е. И. Юревича. М.: Машиностроение, 2007. 360 с.
- [Карпов, Карпова, 2002] Карпов В.Э., Карпова И.П. К вопросу о принципах классификации систем // Информационные технологии. 2002. Т. 2. С. 35–39.
- [Карпов, Павловский, Ройзензон, 2016] Карпов В.Э., Павловский В.Е., Ройзензон Г.В. Многокритериальный подход к определению интеллектуального робота // Пятнадцатая национальная конференция по искусственному интеллекту с международным участием (КИИ-2016), 3-7 октября 2016, Т.3. Смоленск: Универсум, 2016. С. 312–219.
- [Крон, 1972] Крон Г. Исследование сложных систем по частям (диакоптика) / под ред. А.В. Баранова. М.: Наука, 1972. 544 с.
- [Ларичев, 2006] Ларичев О.И. Вербальный анализ решений. М.: Наука, 2006. 181 с.
- [Накано, 1988] Накано Э. Введение в робототехнику. М.: Мир, 1988. 334 с.
- [Павловский, 2014] Павловский В.Е. Интеллектуальная робототехника: современное состояние проблемы. Пленарный доклад // Четырнадцатая национальная конференция по искусственному интеллекту с международным участием (КИИ-2014). Казань: Изд-во РИЦ “Школа”, 2014.
- [Павловский и др., 2016] Павловский В.Е. и др. Информационная система интеллектуального беспилотного автомобиля «АВТОНИВА» // Беспилотные транспортные средства с элементами искусственного интеллекта. Труды семинара. М.: Издательство «Перо», 2016. С. 88–97.
- [Попов, Фирдман, 1976] Попов Э.В., Фирдман Г.Р. Алгоритмические основы интеллектуальных роботов и искусственного интеллекта. М.: Наука, 1976. 456 с.
- [Ройзензон, 2005] Ройзензон Г.В. Способы снижения размерности признакового пространства для описания сложных систем в задачах принятия решений // Новости искусственного интеллекта. 2005. № 1. С. 18–28.

- [**Ройзензон, 2012**] Ройзензон Г.В. Синергетический эффект в принятии решений // Системные исследования. Методологические проблемы. Ежегодник / Под ред. Ю. С. Попкова, В. Н. Садовского, В. И. Тищенко, № 36. 2011-2012. М.: УРСС, 2012. С. 248–272.
- [**Шахинпур, 1990**] Шахинпур М. Курс робототехники. М.: Мир, 1990. 527 с.
- [**Янг, 1979**] Янг Д.Ф. Робототехника. Л.: Машиностроение, 1979. 300 с.
- [**Boudette, 2017**] Boudette N.E. Tesla's Self-Driving System Cleared in Deadly Crash [Электронный ресурс]. URL: [https://www.nytimes.com/2017/01/19/business/tesla-model-s-autopilot-fatal-crash.html?\\_r=0](https://www.nytimes.com/2017/01/19/business/tesla-model-s-autopilot-fatal-crash.html?_r=0) (дата обращения: 10.05.2017).
- [**Simon, Newell, 1958**] Simon H., Newell A. Heuristic problem solving: the next advance in operations research // Oper. Res. 1958. Т. 6. № 1. С. 1–10.

УДК 004.896

## КОНЦЕПТУАЛЬНАЯ И АЛГОРИТМИЧЕСКИЕ МОДЕЛИ СОВМЕСТНОГО ФУНКЦИОНИРОВАНИЯ РОБОТИЗИРОВАННОЙ ПЛАТФОРМЫ И НАБОРА БЛА ПРИ ВЫПОЛНЕНИИ АГРАРНЫХ ОПЕРАЦИЙ

А.Л. Ронжин (*ronzhin@iias.spb.su*)

Санкт-Петербургский институт информатики и  
автоматизации Российской академии наук, Санкт-Петербург

Д.К. Ву (*vuquyenntk@gmail.com*)

В.В. Нгуен (*nguyenvanhvkt@gmail.com*)

О.Я. Соленая (*osolenaya@list.ru*)

Санкт-Петербургский государственный университет  
аэрокосмического приборостроения, Санкт-Петербург

**Аннотация.** Рассмотрены проблемы проектирования подвижной роботизированной платформы, несущей набор беспилотных летательных аппаратов и содержащей контейнер для полезной нагрузки. Обсуждаются вопросы транспортировки роботов и совместной деятельности роботов с различным функционалом при решении аграрных задач. Приведена концептуальная модель конструкции подвижной платформы. Описаны основные этапы алгоритмической модели совместной работы подвижной платформы и набора привезенных беспилотных летательных аппаратов на рабочей территории.<sup>1</sup>

**Ключевые слова:** коллаборативные роботы, агроботы, БЛА, групповое взаимодействие, навигация, манипуляторы.

### Введение

Совместное функционирование групп гетерогенных роботов, а также их доставка/транспортировка к месту непосредственного решения прикладных задач являются одними из основных нерешенных проблем в области робототехники. Существующие решения в области группового управления роботизированных средств в основном решают задачи аддитивного усиления функционала гомогенных роботов. Например, увеличение суммарной тяговой мощности нескольких роботов; захват

---

<sup>1</sup> Работа выполнена при финансовой поддержке РФФ (грант №16-19-00044).

длинномерных объектов несколькими роботами; одновременный контроль распределенных объектов протяженной территории несколькими беспилотными летательными аппаратами (БЛА); ускорение мониторинга и составления картограмм местности за счет использования группы БЛА. Задачи же совместной деятельности роботов с разным функционалом при выполнении параллельных или последовательных операций над объектами рассматриваются еще не так часто и не имеют приемлемых решений.

При проектировании робототехнических средств разработчики в основном ориентируются на функциональные требования, предъявляемые к роботу для решения задач, выполняемых непосредственно в рабочей зоне. Доставка роботов как правило осуществляется не автоматически с привлечением ручного труда. Передвижение с места основного базирования роботов в рабочую зону своим ходом также обычно не предусматривается вследствие ограниченных бортовых энергетических ресурсов и относительно небольших, особенно для наземных робототехнических средств, скоростей передвижения.

В связи с вышеперечисленными двумя проблемами в рамках данного исследования ведется разработка прототипа роботизированной подвижной платформы, использующейся для транспортировки набора БЛА. Среди прочих требований к проектируемой платформе также предъявляется наличие средств контроля, обеспечения энергопитания БЛА для длительной работы и контейнера для полезной нагрузки, например, аграрной продукции или минеральных удобрений. Далее рассмотрим основные направления исследований, проводимые в области аграрной робототехники и группового взаимодействия.

## **1. Анализ современного состояния**

Проблема совместной деятельности группы роботов и операторов сейчас активно обсуждается в России и за рубежом [Ющенко, 2016 и др.; Ronzhin, 2016 et al.]. Во главу угла при этом ставятся вопросы безопасности человеко-машинного взаимодействия, увеличение функциональных возможностей при выполнении операций совместно, за счет устранения влияния человеческого фактора и эмерджентности формируемой системы гетерогенных роботов.

Также робототехника активно внедряется в сельскохозяйственную отрасль, что вызвано во многом социоэкономическими и экологическими факторами. Робототехнические средства становятся особенно востребованными при мелкомасштабном сельском хозяйстве. Беспилотные летательные аппараты сейчас активно начинают использоваться для мониторинга угодий, составления картограмм урожайности земель и планирования зон внесения удобрений. Мультикоптеры, не требуя взлетно-

посадочной полосы, обладают высокой разрешающей способностью и поэтому имеют высокие перспективы широкого применения. В добавление к бортовой видеокамере мультикоптеры могут быть оснащены и другими сенсорными средствами, например: тепловизором, термометром, газовыми датчиками, гидролокаторами, датчиками скорости ветра, датчиками давления, инфракрасными и другими сенсорами.

Отличительной чертой аграрной робототехники является относительно устойчивая регулярность топологии рассадки обрабатываемых культур в отличие от других сфер применения роботов, где обслуживаемые объекты не имеют заранее известных координат и могут перемещаться в пространстве. Сейчас мировой рынок сельскохозяйственных роботов переживает значительный рост [Bechar, 2016 et al.; Holloway, 2014 et al.]. Роботы используются на многих этапах и в различных типах сельского хозяйства: полевые работы, животноводство, производство и сбор продуктов питания [Perez-Ruiz, 2014 et al.; Holloway, 2014 et al.]. Роботизация сельскохозяйственного производства вызвана необходимостью подъема производительности труда; обновление оборудования и техники, избавления работников от физически сложных задач. Наибольший прогресс в роботизации сельскохозяйственного производства в настоящее время виден в области точного земледелия (ТЗ), отличительной чертой которого является локальная дифференциация агротехнических воздействий с учетом пространственной вариабельности почвенных и иных факторов формирования урожая в пределах отдельного сельскохозяйственного поля (агроконтур) [Сидорова, 2012 и др.; Афанасьев, 2016 и др.].

С развитием БЛА начали изучаться задачи их применения не только для мониторинга территорий, но и физического взаимодействия со средой и ее объектами. Технические проблемы проектирования манипулятора для БЛА рассматриваются в ряде современных зарубежных работ [Kondak, 2015 et al.; Baizid, 2017 et al.; Aباunza, 2017 et al.]. Также существует ряд работ, где рассматривается вопрос о взаимодействии или посадке одиночного БЛА на мобильную платформу или судно [Андрейчук, 2016 и др.; Daly, 2015 et al.; Ioannou, 2008 et al.; Sanchez-Lopez, 2014 et al.].

Далее рассмотрим основные направления работ, проводимых в данном исследовании, учитывая полученный ранее задел в области проектирования бортовых вычислителей и робототехнических систем [Гапонов, 2016 и др.; Дашевский, 2015 и др.; Крючков, 2014 и др.; Павлюк, 2016 и др.; Ронжин, 2007 и др.].

## 2. Моделирование конструкции и функций подвижной роботизированной платформы с набором беспилотных летательных аппаратов

Отличительной особенностью разрабатываемой роботизированной платформы является наличие встроенных парковочных мест для нескольких БЛА. На данном этапе исследований приведем концептуальную модель платформы, учитывающую ее оснащение и конструктивные особенности. Основными конструктивными компонентами платформы являются: 1) шасси, обеспечивающая передвижение платформы по сельскохозяйственным угодиям; 2) мультисенсорная система для определения локальных препятствий при перемещении платформы; 3) система энергопитания, обеспечивающая необходимую мощность для потребления самой платформой и набором БЛА при возникновении необходимости их подзарядки; 4) навигационная система, состоящая из двух подсистем, одна из которых контролирует движение платформы между местом базирования и рабочей территорией на основе средств глобальной навигации, а вторая локальная система разворачивается непосредственно на рабочей территории перед началом проведения аграрной операции с использованием радионавигационных БЛА; 5) коммуникационная система, реализующая связь подвижной платформой с БЛА и базовой станций; 6) кузовная часть платформы, содержащая контейнер для полезной нагрузки, а также парковочные места для набора БЛА. Приведенный перечень основных компонентов обеспечивает функционирование платформы в автономных аграрных миссиях и совместную работу БЛА при обработке сельскохозяйственных объектов.

На текущий момент сформировано множество функций, реализуемых подвижной платформой с учетом ее конструктивных особенностей

$$F_P = \{f_1, f_2, \dots, f_8\}, \text{ где:}$$

$f_1$  - движение платформы в двух режимах: 1) движение от места базирования к обрабатываемой территории и назад, 2) движение по сельскохозяйственной территории при выполнении текущих заданий;

$f_2$  - зарядка БЛА с возможной реализацией трех вариантов передачи электрической энергии с использованием системы энергопитания, оснащенной на платформе: 1) контактное соединение аккумулятора БЛА с системой питания платформы; 2) замена аккумулятора БЛА; 3) беспроводная зарядка аккумулятора БЛА;

$f_3$  - контактное взаимодействие платформы с набором БЛА, включая механизм стыковки БЛА и платформы, предусматривающий безопасное передвижение БЛА на борту платформы, посадку и взлет БЛА;

$f_4$  - загрузка БЛА полезными ресурсами (например, минеральные удобрения) из контейнера платформы для распределения по обрабатываемой территории.

$f_5$  - прием грузов (собираемые при уборке плоды, убираемые при чистке камни и другой мусор) с БЛА в контейнер платформы.

$f_6$  - связь подвижной платформы с БЛА и базовой станцией.

$f_7$  - навигация подвижной платформы в глобальной системе координат с использованием системы гетерогенных сенсоров.

$f_8$  - управление планированием и действиями подвижной платформы в кооперации с набором БЛА при решении поставленных аграрных задач.

Для реализации перечисленных функций необходимо применение двух типов БЛА: 1) навигационные БЛА, использующиеся для мониторинга и развертывания системы локальной навигации на обрабатываемой территории; 2) сервисные БЛА, оснащенные захватом или другим инструментом для выполнения операций с полезной нагрузкой.

Далее рассмотрим алгоритмическую модель совместной работы подвижной платформы и набора БЛА, представленную на рисунке 1. Модель укрупненно описывает основные этапы функционирования и совместной работы платформы и БЛА при решении целевой аграрной операции. Первым этапом является планирование маршрута движения и выезд подвижной платформы с находящимся на борту набором БЛА на поле. После достижения платформы заданной точки границы обрабатываемого поля следует этап мониторинга поля, который выполняется с помощью БЛА и бортовых или облачных систем технического зрения с целью составления уточненной цифровой карты местности и при необходимости различных видов картограмм обрабатываемого поля. Затем следует этап развертывания системы локальной навигации. На основе цифровой карты местности посредством БЛА создается сеть опорных станций, размещаемых в точках с известными координатами. Система локальной навигации предназначена для решения основных задач, таких как: обеспечения навигационными данными потребителей (платформы и БЛА) в пределах прямой радиовидимости; формирование навигационного поля для указанных потребителей. Для формирования навигационного поля разворачиваются как минимум три

станции для определения координат потребителей в трехмерном пространстве. Используя созданное навигационное поле, уточненную цифровую карту местности, картограммы БЛА перемещаются по своим траекториям и выполняют целевые задачи, взаимодействуя с передвигающейся платформой. После выполнения целевой задачи БЛА возвращаются к платформе. На следующем этапе платформа с находящимся на борту набором БЛА возвращается на место базирования. Планирование маршрута платформы рассчитывается с учетом текущего места нахождения платформы и системы глобальной навигации.



Рис. 1. Алгоритмическая модель совместной работы подвижной платформы и набора БЛА

Далее рассмотрим несколько процедур из приведенной выше алгоритмической модели. На рисунке 2 приведен блок-схема алгоритма работы платформы и БЛА при мониторинге поля. Вначале проводится определение количества БЛА, необходимых для проведения мониторинга текущей рабочей территории и формирование заданий каждому. Затем производится вылет каждого БЛА с парковочных мест подвижной платформы согласно запланированной траектории полета. В заданных точках БЛА выполняют сбор данных с помощью бесконтактных сенсорных устройств, оснащенных на борту, таких как: видеокамеры, тепловизоры, термометры, инфракрасные и другие сенсоры. Данные могут быть в виде изображений, видео и других показателей окружающей среды. Необработанные данные передаются на платформу с БЛА через беспроводную связь. На следующем этапе проводится обработка данных с целью составления цифровой карты местности, построения картограмм



урожайности, засоренности, агрохимических картограмм и т.д. После мониторинга всей территории в случае недостаточной информации обрабатываемого поля БЛА выполняется повторно сбор данных. После выполнения задачи мониторинга набор БЛА возвращается на платформу.

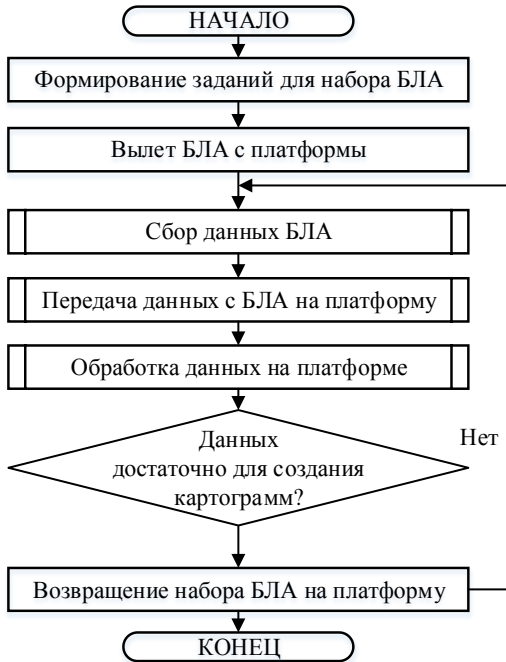


Рис. 2. Блок-схема алгоритма мониторинга поля

На рисунке 3 приведена блок-схема алгоритма развёртывания системы локальной навигации. Основной задачей локальной навигации является определение пространственных координат сервисных БЛА при выполнении заданных операций. Положение сервисных БЛА определяется триангуляционным методом путём измерения расстояний до трёх источников навигационных сигналов, установленных на опорных БЛА. Первым этапом является определение трёх опорных точек, соответствующих трём наиболее удалённым выпуклым вершинам в многоугольнике, описывающим границы поля, вычисленные на предыдущем этапе в ходе мониторинга. После определения координат опорных точек в глобальной навигационной системе производится планирование траекторий БЛА, затем следует этап расстыковки и вылета набора навигационных БЛА с платформы к опорным точкам. В опорных

точках набор БЛА выполняется посадку и фиксацию, чтобы устранить помехи, связанные с ветром, вибрацией и другими внешними факторами. После калибровки и настройки системы локальной навигации, осуществляется назначение координат всем сервисным БЛА запаркованным на подвижной платформе.

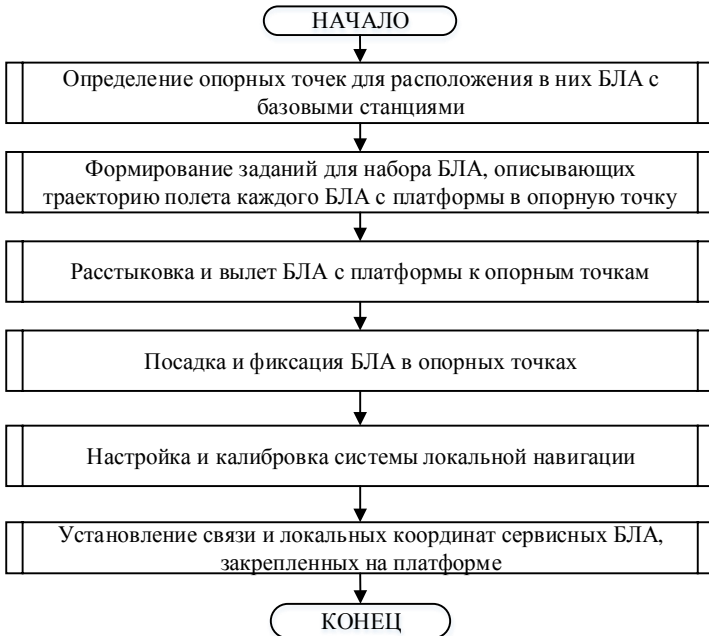


Рис. 3. Блок-схема алгоритма развёртывания системы локальной навигации

В качестве тестовых сценариев совместной работы подвижной платформы и БЛА выбраны задачи аграрной робототехники: 1) мониторинг сельскохозяйственных угодий и построение картограмм с учетом пространственной вариативности; 2) уборка камней и других помех с полей с учетом картограммы засоренности, 3) внесение удобрений с учетом агрохимических картограмм плодородия почвы, планируемой урожайностью культуры, уровня предыдущего урожая, влагообеспеченности, особенностей рельефа полей, агрофизических свойств почвы и других факторов; 4) сбор плодов с учетом картограмм урожайности.

## Заключение

В статье проанализированы основные задачи проектирования подвижной роботизированной платформы, отвечающей за транспортировку и совместную работу с беспилотными летательными аппаратами при выполнении сельскохозяйственных операций. Кратко описана концептуальная модель конструкции платформы и алгоритмическая модель основных этапов взаимодействия платформы с БЛА. Дальнейшее исследование будет посвящено разработке алгоритмических моделей отдельных этапов функционирования платформы и БЛА, а также проектных схем и конструкторских моделей подвижной платформы и ее основных отличительных узлов: контейнера для полезной нагрузки и парковочных мест для БЛА.

## Список литературы

- [**Андрейчук, 2016 и др.**] Андрейчук А.А., Яковлев К.С. Метод разрешения конфликтов при планировании пространственных Труды конференции беспилотные транспортные средства с элементами искусственного интеллекта, 2016, С. 31-40.
- [**Афанасьев, 2016 и др.**] Афанасьев Р.А, Ермолов И.Л. О перспективах роботизации точного земледелия // Мехатроника, автоматизация, управление. 2016. № 12. С. 828–833.
- [**Гапонов, 2016 и др.**] Гапонов В.С., Дашевский В.П., Бизин М.М. Модернизация программно-аппаратного обеспечения модельных сервоприводов для использования в антропоморфных робототехнических комплексах // Доклады Томского государственного университета систем управления и радиоэлектроники. 2016. Т. 19. № 2. С. 41-50.
- [**Дашевский, 2015 и др.**] Дашевский В.П., Бизин М.М. Обзор возможностей бортовых вычислителей на основе SMARK-модулей для робототехнических комплексов // Доклады Томского государственного университета систем управления и радиоэлектроники. 2015. № 3 (37). С. 91-96.
- [**Крючков, 2014 и др.**] Крючков Б.И., Дашевский В.П., Соколов Б.В., Усов В.М. Позиционирование мобильного робота-помощника во внутреннем рабочем пространстве на пилотируемых космических комплексах // Пилотируемые полеты в космос. 2014. № 4 (13). С. 40-56.
- [**Павлюк, 2016 и др.**] Павлюк Н.А., Будков В.Ю., Бизин М.М., Ронжин А.Л. Разработка конструкции узла ноги антропоморфного робота Антарес на основе двухмоторного колена // Известия ЮФУ. Технические науки. 2016. № 1 (174). С. 227-239.
- [**Ронжин, 2007 и др.**] Ронжин А.Л., Карпов А.А., Леонтьева А.Б., Костюченко Б.Е. Разработка многомодального информационного киоска // Труды СПИИРАН. 2007. № 5. С. 227-246.
- [**Сидорова, 2012 и др.**] Сидорова В. А, Жуковский Е. Е, Лекомцев П. В, Якушев В. В. Геоestatистический анализ характеристик почв и урожайности в полевом

- опыте по точному земледелию // *Агрехимия и плодородие почв*. 2012. № 8 . С. 879–888.
- [Ющенко, 2016 и др.]** Ющенко А.С., Ермишин К.В. Коллаборативные мобильные роботы - новый этап развития робототехники. *Экстремальная робототехника*. 2016. Т. 1. № 1. С. 451-459.
- [Abaunza, 2017 et al.]** Abaunza H., Castillo P., Victorino A., Lozano R. Dual Quaternion Modeling and Control of a Quad-rotor Aerial Manipulator. *J. Intell. Robot. Syst.* DOI 10.1007/s10846-017-0519-4
- [Baizid, 2017 et al.]** Baizid K., Giglio G., Pierri F., Trujillo M.A., Antonelli G., Caccavale F., Viguria A., Chiaverini S., Ollero A. Behavioral control of unmanned aerial vehicle manipulator Systems. *Auton. Robot.* 2017, 41, pp. 1203–1220.
- [Bechar, 2016 et al.]** Bechar A., Vigneault C. Agricultural robots for field operations: Concepts and components. *Biosystems Engineering*. 2016. 149. pp. 94–111.
- [Daly, 2015 et al.]** Daly J.M., Ma Y., Waslander S.L. Coordinated landing of a quadrotor on a skid-steered ground vehicle in the presence of time delays. *Auton. Robot.*, 2015, 38, pp. 179–191.
- [Holloway, 2014 et al.]** Holloway L., Bear C., Wilkinson K. Re-capturing bovine life: Robot-cow relationships, freedom and control in dairy farming. *Journal of Rural Studies*. 2014. 33. pp. 131–140.
- [Ioannou, 2008 et al.]** Ioannou S., Dalamagkidis K., Valavanis K.P., Stefanakos E.K. Improving Endurance and Range of a UGV with Gimballed Landing Platform for Launching Small Unmanned Helicopters. *J. Intell. Robot. Syst.* 2008, 53, pp. 399–416.
- [Kondak, 2015 et al.]** Kondak K., Ollero A., Maza I., Krieger K., Albu-Schaeffer A., Schwarzbach M., Laiacker M. Unmanned Aerial Systems Physically Interacting with the Environment: Load Transportation, Deployment, and Aerial Manipulation. K.P. Valavanis, G.J. Vachtsevanos (Eds.), *Handbook of Unmanned Aerial Vehicles*, pp. 2755-2785.
- [Perez-Ruiz, 2014 et al.]** Perez-Ruiz M., Slaughter D.C., Fathallah F.A., Gliever C.J., Miller B.J. Co-robotic intra-row weed control system. *Biosystems Engineering*. 2014. 126. pp. 45–55.
- [Ronzhin, 2016 et al.]** Interactive Collaborative Robotics, A. Ronzhin, G. Rigoll, R. Meshcheryakov (Eds.), LNCS/LNAI, Springer International Publishing Switzerland, Vol. 9812, 2016, 264 p.
- [Sanchez-Lopez, 2014 et al.]** Sanchez-Lopez J.L., Pestana J., Saripall S., Campoy P. An Approach Toward Visual Autonomous Ship Board Landing of a VTOL UAV. *J. Intell. Robot. Syst.* 2014, 74, pp. 113–127.
- [Wolfert, 2017 et al.]** Wolfert S., Ge L., Verdouwa C., Bogaardt M.J. Big Data in Smart Farming – A review. *Agricultural Systems*. 2017. 153. pp. 69–80.

УДК 004.896

## МЕТОД ПОСТРОЕНИЯ СЕМИОТИЧЕСКОЙ СРЕДЫ ФУНКЦИОНИРОВАНИЯ ГРУППОЙ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ

А.А. Кулинич (*kulinich@ipu.ru*)Институт проблем управления РАН, Москва  
НИЦ «Курчатовский институт», Москва

**Аннотация.** Предложен метод построения семиотической среды функционирования группы агентов, основанный на выделении и структуризации возможных классов состояний динамической системы «Группа роботов - Среда» и обозначения этих классов именами-символами. Предложенный метод позволяет группе агентов сформировать согласованное множество имен классов состояний среды функционирования, собственные понятийные системы, что позволяет реализовать их командное поведение для достижения общей цели.<sup>1</sup>

**Ключевые слова:** Роботы, агенты, среда функционирования, имена-символы, семиотическая среда, кооперация.

### Введение

Интеллектуального робота (агента) можно определить перечислением его следующих способностей [Wooldridge, 1994]: 1) *Автономность* – способность действовать самостоятельно; 2) *Реактивность* – способность реагировать на изменение состояний среды функционирования; 3) *Проактивность* – способность проявлять инициативу для достижения поставленных целей; 4) *Социальность* – способность взаимодействовать и договариваться с другими агентами для достижения общей цели.

Реализуются эти способности в рамках различных архитектур интеллектуальных агентов, одна из которых – это BDI-архитектура (*Belief-Desire-Intention*) [Rao, 1995]. BDI архитектура – это ментальная архитектура агента, в рамках которой определяются три составляющие: убеждения (*Belief*) – это знания агента о предметной области, желания

---

<sup>1</sup> Работа выполнена при поддержке грантов РФФИ, проекты: ОФИ\_м № 16-29-04412 и № 15-01-07900 А.

(*Desire*) - это цели агентов и намерения (*Intention*) – это возможные действия агентов для достижения поставленных целей.

Вопросы командного поведения групп интеллектуальных агентов с BDI архитектурой исследовались в работе [Cohen, 1991], в которой вводятся абстрактные принципы (спецификации) организации совместной работы группы агентов на основе их общих намерений, которые выражаются через обязательства, принимаемые агентами при их совместной работе. В другой работе [Grosz, 1996] изложена теория общих планов, основными понятиями которой является понятие группового плана, для выполнения которого агенты должны прийти к соглашению о действиях, которые они будут выполнять, реализуя этот групповой план.

В обзорных работах [Городецкий, 2011] приводятся основные теоретические модели, языки и спецификации командного поведения автономных интеллектуальных агентов. Все эти исследования и разработки в основном направлены на решение конкретных задач группового поведения в рамках классических работ [Cohen, 1991] и [Grosz, 1996]. При этом системные вопросы формирования и функционирования групп автономных агентов и их самоорганизации, как правило, не рассматриваются. Такие вопросы обычно рассматриваются для больших команд агентов с простой реактивной архитектурой (рой, стая) [Карпов, 2016]. Однако, вопросы, например, социального поведения групп интеллектуальных агентов с BDI архитектурой мало изучены.

Сложность моделирования групповой работы интеллектуальных агентов с BDI-архитектурой заключается в том, что у каждого BDI-агента имеется индивидуальная модель мира (*убеждения, знания*), которая определяет его поведение (*намерения*) и может меняться по мере накопления агентом опыта. Элементы этой архитектуры (*убеждения, цели и действия*) выражаются не в числовом, а в символическом виде. По сути, формальное описание всех составляющих архитектуры агента – это его знания о предметной области задаются математическими структурами, элементами которых являются множества понятий, их признаки, значения признаков и множество отношений на всех этих множествах.

Формальное описание знаний агента, несмотря на то что, в настоящее время существует языки описания знаний агентов (агентов) при моделировании кооперативной работы больших групп неоднородных интеллектуальных агентов достаточно трудоемко.

В этих случаях, интерес представляют методы и алгоритмы самостоятельного построения агентами своих моделей знаний о предметной области, на основе опыта работы в этой области и некоторых базовых принципов их построения, представленных, например, в виде метаонтологии (самых общих знаний, правил, методов и т.д.).

В этой работе предлагается метод построения семиотической среды функционирования автономными агентами. Суть метода заключается в том, что в качестве моделей знаний о предметной области используются концептуальные каркасы предметной области, грубо описывающие понятийную структуру предметной области, построение которых не вызывает трудностей [Кулинич, 2014].

## 1. Метаонтология интеллектуального агента

В программной когнитивной архитектуре реактивного агента его когнитивные функции: убеждения, целеполагание, предпочтения агента, реализуются разработчиком агента, как некоторые метазнания, в виде алгоритмов функционирования агента и, фактически, отражают представления разработчика о закономерностях предметной области.

Интеллектуальные агенты с BDI архитектурой по сравнению с реактивными агентами обладают большей свободой выбора поведения агента. Элементом архитектуры, обеспечивающим эту свободу, являются метазнания, представленные в виде метаонтологии, в которой определены самые общие знания о предметной области, модели, правила и др.

Основным элементом метаонтологии агента являются его знания о среде функционирования. Обычно это логическая модель представления знаний, основанная на логике предикатов первого порядка. В этой работе в качестве системной модели знаний о среде функционирования будет рассмотрена динамическая модель многоагентной системы [Каляев и др., 2009]. Здесь группа роботов  $R_i, i=1, \dots, N$ , каждый из которых характеризуется вектором состояний  $R_i=(r_{i1}, \dots, r_{in})$  (вектором значений его характеристик) находится в среде  $E$ , которая также характеризуется вектором состояния,  $E=(e_1, \dots, e_n)$ . Пара векторов  $S=\langle R, E \rangle$  характеризует состояние системы «группа роботов – среда» в пространстве состояний ( $SS$ ), которое определяется как прямое произведение параметров всех агентов  $R_i$ , объектов и состояний среды ( $E$ ), т.е.,  $SS=\prod_i R_i \times E$ .

Пространство состояний  $SS$  может интерпретироваться как признаковое семантическое пространство, в котором объекты реального мира, различные ситуации представляются как понятия. В семантическом пространстве объекты (ситуации), по сути, денотаты, определяются именами и векторами значений признаков, определяющие их содержание (смысл) и могут интерпретироваться как знаки-символы в определении Г. Фреге [Бирюков, 1960]. Знак определяется тройкой: имя, смысл и значение знака. Имя – это символ (образ, икона) обозначающее объект реального мира, смысл определяет свойства этого объекта, а значение – это и есть сам объект – денотат. В семантических пространствах реальный объект

(денотат) имеет имя (символ) и представляется в признаковом пространстве как точка, координаты которой определяют значение признаков (смысл).

Определим множество  $F=\{f_i\}$  как множество всех свойств (признаков) всех агентов и объектов, а  $Z_i$  как возможные значения этих свойств. Тогда среда функционирования определится как прямое произведение значений, т.е.  $SF=\times_i Z_i$ , а агенты и объекта как точки в этом пространстве.

В рамках этой системной модели знаний метаонтология каждого агента включает: основные понятия предметной области  $D=\{d^H\}$ ; свойства (признаки) этих понятий и их возможные значения ( $Z_i$ ) -  $F=\{f_i, Z_i\}$ ; тип шкалы значений признаков {числовые, качественные, номинальные, отношений}; типы отношений, заданных на множестве понятий предметной области и их признаков (это отношения «Род-Вид», («Класс-Подкласс»), «Причинно-следственные» и др.); закономерности предметной области, представленные как отношения «Род-Вид» множестве понятий предметной области, т.е.  $R\subseteq D\times D$ , и причинно-следственные отношения на множестве значений признаков, т.е.  $W\subseteq\times_i Z_i$ .

Метаонтология определяет основные элементы модели знаний о предметной области каждого агента из которых агентами могут быть построены их индивидуальные модели предметной области, в зависимости от их опыта работы и индивидуальных возможностей.

Для реализации социальной способности интеллектуальных агентов (их коммуникации), каждый агент строит свою модель знаний о предметной области, опираясь на правила, определенные в метаонтологии. В этой работе эти правила задает качественный концептуальный каркас этой области [Кулинич, 2014], который, по сути, является идеализированной онтологией этой предметной области.

Формально качественный концептуальный каркас – это частично упорядоченное множество имен классов состояний агента (робота) в среде функционирования, ( $\{d^H\}, \leq$ ). При этом каждое имя ( $d^H$ ) определяет подпространство  $SS(d^H)\subseteq\times_i Z_i$ , определяющее этот класс состояний.

Так как, модель знаний любого агента является частью качественного концептуального каркаса предметной области, то и коммуникация между агентами будет осуществляться с помощью искусственных имен ( $d^H$ ) классов состояний концептуального каркаса и будет понятна всем агентам. Концептуальный каркас предметной области в дальнейшем будем называть семиотической средой функционирования.



## 2. Алгоритм построения семиотической среды функционирования

Пусть названия признаков (имена) множества признаков среды функционирования  $\{f_i\} \in F$  агентов являются элементами метаонтологии и заложены в память агентов и определены ресурсы, которые позволяют агенту изменять значения признаков среды функционирования. Под ресурсом агента понимаем возможность агента изменять значения своих свойств (признаков), свойств других агентов или объектов.

То есть, ресурс по ( $j$ -у свойству) признаку  $f_{ij}$   $i$ -о агента это пара  $(f_{ij}; Z_{ij}^R)$ , где  $Z_{ij}^R = \{z_{ie}, \dots, z_{iq}\} \subseteq Z_j$  – строго упорядоченное множество значений. Если каждый агент может изменить значения нескольких своих свойств, то тогда его ресурсы будут заданы вектором, например, для первого агента – это вектор:  $(f_{11}; Z_{11}^R), \dots, (f_{1m}; Z_{1m}^R)$ .

Считаем, каждый из  $N$  агентов знает о своих ресурсах:

$$\begin{aligned} &(f_{11}; Z_{11}^R), \dots, (f_{1m}; Z_{1m}^R); \\ &\dots\dots\dots \\ &(f_{n1}; Z_{n1}^R), \dots, (f_{nm}; Z_{nm}^R). \end{aligned} \tag{1}$$

Задача заключается в том, чтобы по имеющейся у агентов информации о ресурсах построить концептуальный каркас среды функционирования  $(\{d^H\}, \leq)$ , определить множество имен классов состояний  $(\{d^H\})$ , соответствующие им подпространства  $SS(d^H)$ , т.е. определить семиотическую среду функционирования и свою понятийную систему в терминах искусственных имен этой среды.

Эта задача решается алгоритмически и включает ряд шагов, основанных на обмене информацией через «доску объявлений».

Алгоритм включает следующие шаги, выполняемые каждым агентом:

1. Передача информации о собственных ресурсах на «доску объявлений» для определения пространства их возможного совместного функционирования;
2. Определение базового класса состояний концептуального каркаса среды функционирования;
3. Построение концептуального каркаса среды функционирования и символизация классов состояний (определение искусственных имен);
4. Определение агентами собственной понятийной системы в терминах искусственных имен семиотической среды функционирования.

На первом шаге алгоритма агенты помещают информацию о своих ресурсах на «доску объявлений». Для определения пространства их возможного совместного функционирования агенты, по имеющейся информации (1) о ресурсах каждый агент объединяет эти значения. Т.е.

$$(f_{\cup 1}; \bigcup_{i=1}^n Z_{i1}^R), \dots, (f_{\cup m}; \bigcup_{i=1}^n Z_{im}^R). \quad (2)$$

**Определение 1.** Средой возможного кооперативного взаимодействия агентов будем называть прямое произведение объединения их ресурсов,

т.е.  $\times \bigcup_{j=1}^n Z_{ij}^R$ .

На **втором шаге** каждый агент определяет базовый класс состояний концептуального каркаса среды. Для этого, агенты определяют границы (максимальное и минимальное значение) одноименных ресурсов и находят среднее значение для каждого ресурса, полученных от всех других агентов.

Далее мы будем считать, что существует отображение значений объединенных ресурсов, представленных в порядковых или номинальных

шкалах, на отрезок числовой оси  $[0,1]$ , т.е.  $\varphi: (\bigcup_{i=1}^n Z_{ij}^R) \rightarrow [0,1]$ .

Тогда, среднее значение каждого из объединенных ресурсов агенты определяют по очевидной формуле:

$$z_{ij} = (\max(\varphi(\bigcup_{i=1}^n Z_{ij}^R) - \min(\varphi(\bigcup_{i=1}^n Z_{ij}^R)))/2) \quad (3)$$

Границы интервалов значений признаков базового класса каждый агент определяет как положительное и отрицательное отклонение от их средних значений:  $\Delta i = z_{ij} \pm \varepsilon$ , где  $\varepsilon = (10-15\%) z_{ij}$ , а сам базовый класс как прямое произведение, полученных интервалов, т.е.

$$SS(d_0) = \times i \Delta i. \quad (4)$$

Отметим, что поскольку все агенты работают по одному и тому же алгоритму определения базового класса состояний, то базовый класс состояний у всех агентов будет одинаков.

В качестве иллюстрации определения базового класса, на рисунке 1 показан базовый класс состояний среды функционирования для двух признаков  $X \in [0,1]$  и  $Y \in [0,1]$ .

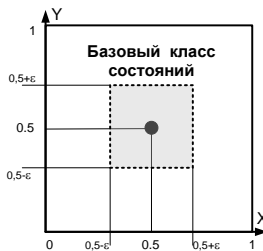


Рис. 1. Базовый класс состояний среды функционирования

Следующий шаг алгоритма – **построение и символизация концептуального каркаса среды функционирования**. Базовый класс состояний – это «идеальная точка» для всех агентов, относительно которой будет строиться концептуальный каркас, и символизироваться среда функционирования.

Общий принцип символизации среды функционирования заключается в качественном кодировании значений признаков, в виде множества символов, например,  $Z_i^O = \{0, 1, 2\}$ . Качественные значения обозначают интервалы значений признаков и определяются по правилу (5).

$$Z_i^O = \begin{cases} 0, & \text{если } z_i \in [z_{ie}^b; z_{iq}^b] \\ 1, & \text{если } \max(Z_{ij}^{R \cup}) \geq z_i > z_{iq}^b; \\ 2, & \text{если } \min(Z_{ij}^{R \cup}) \leq z_i < z_{ie}^b. \end{cases} \quad (5)$$

Т.е., значения признака  $z_i \in Z_i$  попадающие в интервал значений признака базового понятия кодируются цифрой «0», превышающее значение базового интервала, цифрой «1», а если значение признака меньше базового значения, то, соответственно, «2».

Построение концептуального каркаса заключается в определении имен классов состояний среды функционирования и подпространств среды функционирования и соответствующих этим именам.

**Определение 2.** Именем класса состояний качественной среды функционирования будем называть конкатенацию подмножеств прямого произведения качественных значений всех признаков среды функционирования, т.е.  $d^{name}$ , где  $name = K(\{*\})$ ,  $K$  – операция конкатенации,  $\{*\} \in \times_i Z_i^O$ .

Например, есть два признака, качественные значения которых соответственно,  $Z_1^O = \{0, 1, 2\}$  и  $Z_2^O = \{0, 1, 2\}$ . Их прямое произведение ( $Z_1^O \times Z_2^O$ ) равно:  $\{\{0,0\}; \{0,1\}; \{0,2\}; \{1,0\}; \{1,1\}; \{1,2\}; \{2,0\}; \{2,1\}; \{2,2\}\}$ , а конкатенация подмножеств этого произведения равна:  $\{\langle 00 \rangle; \langle 01 \rangle; \langle 02 \rangle; \langle 10 \rangle; \langle 11 \rangle; \langle 12 \rangle; \langle 20 \rangle; \langle 21 \rangle; \langle 22 \rangle\}$ . Имена классов состояний, соответственно, равны:  $\{d^{00}; d^{01}; d^{02}; d^{10}; d^{11}; d^{12}; d^{20}; d^{21}; d^{22}\}$ .

Элементы прямого произведения качественных значений  $Z_i^O$  признаков среды функционирования однозначно определяют подпространства, определяющие обобщенные классы состояний среды функционирования. Действительно, подмножество  $\{0,0\}$  согласно правилу (3) определяет базовые интервалы по первому  $[z_{1e}^b; z_{1q}^b]$  и по второму признаку  $[z_{2e}^b; z_{2q}^b]$ , а их произведение,  $SS(d^{00}) = [z_{1e}^b; z_{1q}^b] \times [z_{2e}^b; z_{2q}^b]$  дает нам базовый класс состояний. Для случаев, когда качественное значение одного из признаков принимает 1, при определении подпространства в качестве интервала

берем интервал  $[z_{ie}^b; \max(\bigcup_{i=1}^n z_{ij}^R)]$ , а если 2, т.е. значение меньше базового интервала, то  $[\min(\bigcup_{i=1}^n z_{ij}^R); z_{iq}^b]$ .

Для подмножества  $\{0,1\}$  подпространство, определяющее класс состояний с именем  $d^{01}$  равно  $SS(d^{01})=[z_{ie}^b; z_{iq}^b] \times [z_{ie}^b; \max(\bigcup_{i=1}^n z_{ij}^R)]$ .

Таким образом, искусственные имена классов состояний определяют подпространства классов состояний среды функционирования и образуют частично упорядоченное множество, т.е. концептуальный каркас среды функционирования.

**Определение 3.** Качественным концептуальным каркасом среды функционирования агентов будем называть частично упорядоченное множество имен классов состояний среды функционирования  $(\{d^H\}, \leq)$ , однозначно определяющие подпространства этой среды  $SS(d^H)$ .

Таким образом, агенты, обмениваясь информацией о своих ресурсах, могут построить концептуальный каркас среды функционирования, опираясь на общие правила его структуризации и символизации.

Следующий шаг – построение агентами собственной **понятийной системы** о среде функционирования.

**Определение 4.** Понятийной системой агента будем называть частично упорядоченное множество имен классов состояний полученное прямым произведением качественных доменов всех его ресурсов, т.е.  $(\{d_i^H\}, \leq)$ , где  $d_i^H \Leftrightarrow SS(d^H)$ ,  $SS(d^H) \subseteq \times_j Z_{ij}^R$ .

### 3. Пример

Пусть агенты *Агент 1* и *Агент 2* знают, что они находятся на плоскости  $\{X, Y\}$  и могут изменить свое положение, изменив свои координаты и состояние среды функционирования.

У агента 1 есть ресурс, позволяющий ему перемещаться только по координате X, (горизонтально):  $X_1 = \{x_{1i}\}$ ,  $x_{1i} \in [0,1]$ ,  $Y_1 = y_{1j}$ ,  $y_{1j} = \text{const} = 0,5$ . У агента 2 есть ресурс, позволяющий ему перемещаться только по координате Y, (вертикально):  $Y_2 = \{y_{2j}\}$ ,  $y_{2j} \in [0,1]$ ,  $X_2 = x_{2i}$ ,  $x_{2i} = \text{const} = 0$  (Рис. 2).

Объединенные ресурсы агентов:  $Y_1 \cup Y_2 = [0,1]$  и  $X_1 \cup X_2 = [0,1]$ , т.е. ресурсы агентов это вектор:  $(f_{\mathcal{Y}}; [0,1], f_{\mathcal{X}}; [0,1])$ , а среда возможного совместного функционирования агентов это произведение:  $[0,1] \times [0,1]$ .

Агенты определили базовый класс состояний по формулам 3 и 4, показанный на рис. 1 и символизировали среду функционирования согласно правилу (5). Пример символизации приведен в определении 2 и показан на рис. 2.

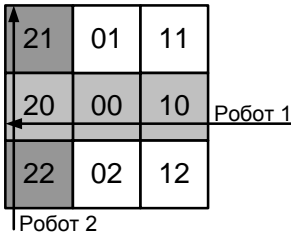


Рис. 2. Понятийные системы агента 1 и агента 2

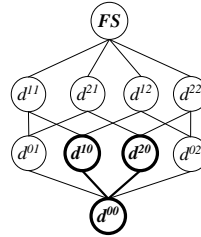


Рис. 3. Диаграмма Хассе понятийной системы агента 1

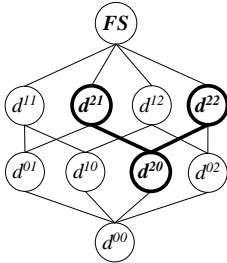


Рис. 4. Диаграмма Хассе понятийной системы агента 1

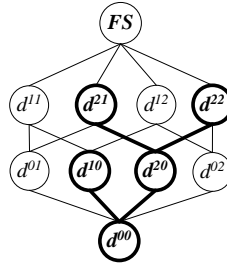


Рис. 5. Диаграмма Хассе области совместного функционирования агентов 1 и 2

Далее агенты определяют собственные понятийные системы. Агент 1 по координате Y может изменять значения этого параметра в пределах базового класса, т.е.  $Z_Y^{Q1} = \{0\}$ , а по координате X он может изменять значение этого параметра больше и меньше значения базового класса, т.е. качественные значения его ресурсов равны:  $Z_X^{Q1} = \{0,1,2\}$ . Тогда знания этого агента о классах среды функционирования определим:  $Z_X^{Q1} \times Z_Y^{Q1} = \{0,1,2\} \times \{0\} = \{00, 10, 20\}$ , или в именах классов состояний  $\{d^{00}, d^{10}, d^{20}\}$ .

Агент 2 по координате X может находиться в состояниях, когда его значения меньше значения X базового класса,  $Z_X^{Q2} = \{2\}$ , а по координате Y ресурсы позволяют ему переходить в классы  $\{0,1,2\}$ , т.е.  $Z_Y^{Q2} = \{0,1,2\}$ . Тогда его знания определяться так:  $Z_X^{Q2} \times Z_Y^{Q2} = \{2\} \times \{0,1,2\} = \{20, 21, 22\}$ , или в именах классов состояний  $\{d^{20}, d^{21}, d^{22}\}$ .

Символизированные понятийные системы агента (агента) 1 и 2 показаны на рис. 2. Диаграммы Хассе понятийных систем агента 1 и 2 показаны на рис. 3 и рис. 4. Диаграмма Хассе области возможного совместного функционирования показана на рис. 5.

## Заключение

В работе предложен метод построения качественной семиотической среды функционирования группой агентов, основанный на выделении и структуризации возможных классов состояний динамической системы «Группа роботов - Среда». Классы состояний обозначены именами-символами и образуют частично упорядоченное множество имен, названное семиотической средой. Предложен алгоритм, позволяющий группе агентов сформировать и согласовать множество искусственных имен классов состояний среды и построить собственные понятийные системы - знания об этой среде в терминах этих имен. Предложенные метод и алгоритм являются частью исследований [Кулинич, 2016], направленных на моделирование социального поведения групп агентов с BDI-архитектурой, функционирующих в семиотической среде.

## Список литературы

- [Wooldridge, 1994] M.Wooldridge and N.R.Jennings. Agent Theories, Architectures, and Languages: A Survey. In: Intelligent Agents. ECAI-94 Workshop on Agent Theories, Architecture and Languages. Amsterdam, The Netherlands, August 8-9, 1994, Proceedings. Springer Verlag: 3-39, 1994.
- [Rao, 1995] Rao A.S., Georgeff M.P. BDI Agents: From Theory to Practice // Proc. First International Conference on Multi-Agent Systems (ed. V.Lesser). – AAAI Press/The MIT Press, 1995. – P. 312–319.
- [Cohen, 1991] Cohen P., Levesque H.J. Teamwork. *Nous*, 25(4), (1991) Special Issue on Cognitive Science and Artificial Intelligence, P. 487-512.
- [Grosz, 1996] Grosz B., Kraus S. Collaborative Plans for Complex Group Actions // *Artificial Intelligence*. – 1996. – №86. – P. 269–358.
- [Городецкий, 2011] Городецкий В.И. Теория, модели, инфраструктуры и языки спецификации командного поведения автономных агентов. Обзор (Часть 1, Часть 2) // *Искусственный интеллект и принятие решений*. – 2011. – №2, №3 – С. 19–30, С. 34–47.
- [Карпов, 2016] Карпов В. Э. Модели социального поведения в групповой робототехнике / Управление большими системами. Выпуск 59. М.: ИПУ РАН, 2016. С.165-232.
- [Кулинич, 2014] Кулинич А.А. Концептуальные каркасы онтологий слабо структурированных предметных областей // *Искусственный интеллект и принятие решений*. 2014. - № 4. - С. 31-41
- [Каляев и др., 2009] Каляев И.А., Гайдук А.Р., Капустян С.Г. Модели и алгоритмы коллективного управления в группах роботов / М.: Физматлит, 2009.-279 с.
- [Бирюков, 1960] Бирюков Б.В. Теория смысла Готлоба Фреге / В кн.: Применение логики в науке и технике. – М.: Изд-во АН СССР, 1960. – С. 502–555.
- [Кулинич, 2016] Кулинич А.А. Модель командной работы агентов с BDI архитектурой. 15-я национальная конференция по искусственному интеллекту

КИИ-2016 (3-7 октября 2016 г., г. Смоленск, Россия): Труды конференции. В 3-х томах. Т.2. – Смоленск: Универсум, 2016. - с. 344-352.

УДК 004.82

## ПЛАНИРОВАНИЕ ДЕЙСТВИЙ КОАЛИЦИЕЙ АГЕНТОВ: КОММУНИКАЦИОННЫЙ АСПЕКТ

Г. А. Киселёв (*kiselev@isa.ru*)

Федеральный Исследовательский Центр “Информатика и управление” Российской Академии Наук, Москва

А. И. Панов (*pan@isa.ru*)

Федеральный Исследовательский Центр “Информатика и управление” Российской Академии Наук, Москва

**Аннотация.** В статье приводится аналитический обзор работ по многоагентному планированию, включающий исследования, связанные с построением координационных сообщений агентов, и исследования, основанные на когнитивном подходе к планированию действий гетерогенными агентами. В заключении приведен метод составления сообщений агентами на основе психологически правдоподобной модели представления знаний.<sup>1</sup>

**Ключевые слова:** многоагентное планирование, многоагентные системы, компьютерное когнитивное моделирование, коммуникации агентов, знаковая картина мира

### Введение

В статье приведен аналитический обзор работ по теме многоагентного планирования с использованием сообщений для координации действий между агентами и предложен алгоритм многоагентного гетерогенного планирования MultiMAP. Существуют различные протоколы организации взаимодействия агентов, такие как аукционы, договорные сети, протоколы с использованием агентов-посредников и другие [ГІРА; Тарасов, 2002]. Ключевыми критериями для установления взаимодействия агентов являются ситуации наличия совместных целей агентов, нехватки ресурсов для самостоятельного достижения цели, неспособность агента решить поставленную задачу самостоятельно или принятые агентом обязательства по оказанию услуг другому агенту. При наличии совместных целей у агентов возникают ситуации, в которых становится выгодно вступать в

---

<sup>1</sup> Работа выполнена при финансовой поддержке РФФИ (проекты № 16-37-60055 мол\_a\_дк и 17-07-00281 а).



кооперацию и действовать совместно, дополняя действия друг друга. Если у агента ощущается нехватка одного вида ресурсов, но присутствует избыток другого вида, агенты способны обмениваться ресурсами для удовлетворения собственных потребностей. Далее в обзоре будут рассмотрены работы, посвященные многоагентному планированию с использованием различных способов коммуникации между агентами, даны основные характеристики численного и когнитивного подходов. В заключение предлагается метод, позволяющий преодолеть основные недостатки рассмотренных в обзоре алгоритмов.

## 1. Планирование в многоагентной системе

Следуя [Granatyr, 2015] выделим 3 основных типа многоагентных систем: системы, основанные на когнитивном подходе, на численном и на гибридном. Когнитивный подход связан с моделированием убеждений и внутренних состояниями агентов. Вводится понятие доверия, которое является степенью готовности агента взаимодействовать с другим агентом. В когнитивном подходе к многоагентному планированию доверие используется в качестве шкалы оценки вероятности установления взаимодействия с другим когнитивным агентом. В качестве примера такого подхода можно привести архитектуру BDI [Sabater, 2005]. Первая когнитивная модель была предложена [Castelfranchi, 2001], где внутреннее состояние агента состояло из когнитивных элементов, таких как цель, способность, компетентность и зависимость. Другим примером является модель, предложенная [Neville, 2004], которая основана на социальной теории доверия, репутации, рекомендации и обучения через непосредственный опыт, интегрированный в механизм рассуждения агента. Модель [Carter, 2003] учитывает сочетание чувства собственного достоинства, репутации и фамильярности среди агентов, в то время как [Zhang, 2007] утверждают, что близость напрямую влияет на доверие. Когнитивный подход позволяет создавать многоагентные системы, в которых взаимоотношения между агентами выстраиваются с использованием психологических подходов, удовлетворяющих антропоморфной модели поведения. В рамках задач многоагентного планирования этот подход позволяет в одном формализме описывать действия интеллектуальных искусственных агентов и действия человека.

Численный подход не использует когнитивные представления и основывается на реактивной архитектуре. Этот подход основан на численном агрегировании прошлых взаимодействий и представляет набор субъективных вероятностей того, что агенты будут правильно выполнять поставленную задачу. В рамках численного подхода упрощается процесс расширения системы новыми агентами и унифицируются правила

построения взаимоотношений между агентами, но теряется возможность использования интеллектуальных агентов в качестве помощников при реализации многоагентного процесса планирования, участниками которого могут выступать люди.

Также стоит отметить гибридные подходы, которые используют обе парадигмы, такие как [Matt, 2010] [Parsons, 2011]. Эти модели являются когнитивными, потому что они основаны на убеждениях и численными, потому что они используют численные агрегации. Подходы такого типа позволяют использовать возможности как численного, так и когнитивного подходов, позволяя реализовывать взаимодействия между искусственным агентом и человеком, расширяя базовые способы взаимодействия за счёт представления знаний, включающее представления о других агентах. Также, в гибридных подходах к многоагентному планированию существует возможность увеличения числа участников коллаборации за счет агентов окружающей среды, минимизируя затраты на установление взаимодействия с ними из-за построения унифицированного механизма взаимодействий, например, на основе речевых актов естественного языка.

Рассмотрим представленные подходы к многоагентному планированию подробнее и отметим сильные и слабые стороны различных алгоритмов.

## 2. Численный подход

В статье [Jones, 2006] рассматривается подход, основанный на отсутствии возможности агентов самостоятельно достигнуть поставленных целей. Агенты, представленные роботами Segway и Pioneer, взаимодействуют при решении задачи поиска ресурсов, дополняя способности друг друга, выполняя задачи визуального нахождения требуемого объекта и построения карты местности, в которой происходит их действие. В статье была реализована многоуровневая архитектура, позволяющая абстрагироваться от реализации механизмов рассылки сообщений другим искусственным интеллектуальным агентам и позволяющая использовать единый унифицированный механизм переговоров для любого из участников гетерогенной группы.

Агенты в коллективах не всегда способны и желают раскрывать свои личные данные и возможности другим агентам, участвуя в аукционах, либо любых других публичных актах создания коалиции. Для реализации такого подхода в рамках процесса многоагентного планирования может использоваться модель, предложенная в статье [Jakubův, 2015]. В работе описывается подход к планированию с использованием многоагентной версии языка PDDL. Факты задачи разбиваются каждым агентом на три подгруппы: публичные, внутренние и релевантные. Для каждого агента создается локальная проблема планирования, которая состоит из общей

проблемы планирования составленной из фактов, релевантных для агента. Множество решений задачи планирования представляется в виде недетерминированного конечного автомата. В статье описывается построение полных и частичных конечных планировочных автоматов. В процессе планирования для каждого конечного планировочного автомата (PSM) строится его публичная проекция, которая представляет публичную проекцию решений, составленную из публичных фактов. Следующим шагом, после построения PSM для каждой локальной проблемы агентов является процесс поиска их пересечений. После нахождения пересечения всех PSM, любое решение, входящее в это множество будет являться публичным решением исходной задачи планирования. Агенты посылают друг друга сообщения, в которых указывают публичную проекцию своего PSM. Каждый из агентов осуществляет процесс планирования в рамках соображений о приоритизации действий, который заключается в предпочтении внутренних действий агента публичным и в предпочтении личных публичных действий публичным действиям других агентов.

В статье [Guo, 2016] описан процесс многоагентного планирования, включающий разделение агентов по гомогенным группам. Окружающая агентов среда разделяется на области, у каждого агента существуют представления о времени перехода в каждую из областей и о действиях, которые он может в ней совершить. Для коммуникации агентов используется двухслойная сеть коммуникаций и создаются группы коммуникаций, которые позволяют выбирать агента-координатора внутри группы агентов и предоставляют его координаты всем агентам других групп. На первом слое агент коммуницирует только со своей группой и может отправить сообщение любому из агентов своей группы, на втором слое агент может отправить сообщение любому из ближайших агентов. Все действия агентов разделяются на три группы: действия личные для агента, коллаборативные и вспомогательные. Агент может инициировать личные и коллаборативные действия. Процесс планирования осуществляется с использованием недетерминированных конечных автоматов. В начальный момент времени агент самостоятельно пробует достичь всех целевых состояний, если у него это не получается, то он извещает всех остальных агентов о требуемой помощи. Были разработаны различные типы сообщений, включающие сообщения о запросе помощи, ответы на сообщения такого типа и запросы помощи к группам других агентов. В группах гомогенных агентов существует возможность обмена частями полученных планов с другим агентом, для чего были описаны правила обмена, включающие последовательность выполнения действий агентами. Описанный в статье протокол построения многоагентных коммуникаций позволяет искусственным интеллектуальным агентам разделять обязанности при совершении действий в процессе многоагентного

планирования, а также упрощает процесс поиска новых агентов, унифицируя алгоритм увеличения численности группы гомогенных агентов за счёт уведомления о поступлении нового агента только ближайших агентов и агента-координатора группы. Использование численного подхода к многоагентному планированию позволяет увеличивать коллектив агентов-спутников гетерогенными агентами, унифицируя механизмы распределения ролей между спутниками и адаптацию полученных сообщений. Один агент может иметь одну или обе роли и переключаться между ними во время выполнения действий в зависимости от ситуации, с которой он сталкивается. Агенты, имеющие ограниченную роль, управляют ограничениями, которые должны быть удовлетворены, в то время как агенты, имеющие служебную роль используются для помощи агентам с ограниченной ролью.

Численный подход к многоагентному планированию позволяет выстраивать чёткую взаимосвязь между агентами, грамотно распределять роли между агентами, минимизируя ресурсы, потребляемые агентами в процессе выполнения плана, но не позволяет создавать условия для установления взаимодействия между искусственным агентом и человеком, что в свою очередь можно осуществить с помощью когнитивного подхода к планированию.

### **3. Когнитивный подход**

В рамках многоагентного когнитивного планирования агенты формируют коалиции для совершения требуемых действий, основываясь на своих убеждениях и предпочтениях. Убеждения агента могут формироваться на основе взаимодействий с другим агентом, социальных норм, принятых в обществе, в котором находится агент, либо на основе психологического состояния самого агента. Как утверждалось ранее доверие является немаловажным фактором для построения многоагентных взаимоотношений. В рамках многоагентного планирования понятие доверия используется в качестве оценочного критерия отбора агентов для построения совместного плана работ. В статье [Granatyr, 2015] утверждается, что непосредственное взаимодействие важно для установки взаимного доверия, но не необходимо. Желая купить что-то у агента «В», агент «А» исследует прецеденты действия агента «В», составляя характеристику его благонадежности. Помимо выбора, основанного на прецедентах у агента может быть список предубеждений, следуя которому он совершает выбор об установлении взаимодействий с другим агентом, либо об их отсутствии. Многоагентная система может иметь список правил, являющихся социальными нормами, схожими для всех агентов. Примером многоагентной системы с правилами поведения агентов

является модель, описанная в статье [Carter, 2002], в которой надежность системы оценивается с помощью четырех основных правил: интерактивность, предоставление контента, обратная связь и долговечность.

В статье [Roy, 2005] более детально представлены механизмы рассуждений когнитивных агентов и описываются внутренние представления окружающей среды агентом, как о наборе объектов со свойствами, которые участвуют в событиях, вызванных агентами. Были сформулированы свойства, которыми должен обладать язык агентов:

1. Унификация репрезентативных примитивов: объекты, свойства, события и ситуации должны быть построены из одного и того же набора базовых примитивов. Это требование необходимо для перевода убеждений агента о конкретных объектах и ситуациях в ожидания результатов действий, связанных с этими объектами.

2. Кросс-модальная переносимость: информация, полученная от восприятия и языка, должна быть интерпретирована в общую репрезентативную форму для проектирования агентов, способных дискутировать о том, что они наблюдают и делают.

3. Интегрированное пространство действий: двигательные акты (например, склонение к разрешению визуальной неопределенности) и речевые акты (например, формулировка вопроса для разрешения визуальной двусмысленности - «это чашка или банка?») должны выражаться в едином интегрированном пространстве действий, чтобы агент мог совместно планировать свои действия с речевыми и двигательными актами для достижения целей.

В статье используются указатели на объекты, которые были названы знаками. Знаки могут состоять из других знаков. Знаки интерпретируются информационными структурами, которые модифицируются перцептивной информацией и называются схемами. Взаимодействие между схемами и окружающей средой опосредуются восприятием и двигательным действием. Описываются два вида схем:

1. С проекции сенсоров выстраивается аналоговый знак объекта.

2. Схема — трансформатор. С помощью схемы такого типа формируется суждение об объекте исходя из его признаков.

В статье было сформулировано понятие категориальных убеждений. Категориальные убеждения рассматриваются как убеждения относительно ответов на вербальные вопросы, которые можно задавать об аналоговых наблюдениях. Каждый раз, когда агент выполняет примитив действия, результат является категориальным знаком о мире, на который агент воздействовал. Также используются индексные знаки — знаки, отвечающие за пространственное расположение агента и объектов. Схемы учитывают цели в отношении объектов и предоставляют прогнозы

результатов последовательностей действий, которые получают желаемый результат. Язык состоит из речевых актов, представленных в статье каноническими целевыми знаками. Как и все знаки, целевые знаки являются физическими шаблонами, которые обозначают что-то для кого-то. Ссылка от целевого знака к означенному устанавливается соглашениями, согласованными сообществом агентов целевого знака. Речевые акты состоят из лексических единиц, составленных с помощью грамматик. В статье используются описания и директивы, являющиеся утверждениями о состоянии мира и запросами на действия. Речевые акты могут быть как описательными, так и директивными. Описания и директивы составляются из расчёта влияния на ощущение окружающей среды другими агентами и используют способности агента предсказывать действия других агентов. Директивные речевые акты интерпретируются в схемы целей, которые агент может выбрать для достижения в процессе планирования действий. Описательные речевые акты интерпретируются в экзистенциальные убеждения, представленные через схемы, которые совместимы посредством восприятия и действия. В работе был описан подход к представлению знаний об окружающем мире интеллектуальным агентом, но не было выделено личностной составляющей агента в обществе других агентов. Преднамеренные знаки, используемые в качестве речевых актов, не отражают пожеланий агента касательно внутренних состояний других агентов коммуникации, но отражают описание среды и желаемые действия агентов. Подход описанный в статье является когнитивным и может быть использован для реализации процесса многоагентного планирования с использованием не только искусственных агентов, но и человека.

В статье [Schrodt, 2017] рассматривается интеллектуальная когнитивная архитектура, которая позволяет извлекать символические концептуальные структуры из непрерывного сенсомоторного опыта. В результате формируются развивающиеся правила и семантические символы для описания сенсомоторных действий. Правила последовательных действий приобретаются из опыта действия. Описываемая архитектура имеет название SEMLINCS. Был рассмотрен пример на основе известной игры MARIO, где в процессе анализа действий агента архитектура выдвигала предположения по поводу последствий действий, основываясь на полученном ранее опыте. Общение с пользователем игры формируется на основе контекстно-свободной грамматики с использованием схем действий, что позволяет формулировать содержательные советы относительно прогнозируемых результатов действий. Пользователь может формировать не только запросы на реализацию определённого действия, но и формулировать сложную цель, состоящую из последовательности действий. В систему были интегрированы средства распознавания речи

CMU Sphinx-4 [SPHINX-4], основанные на скрытых марковских моделях. Во время распознавания речи каждое распознанное предложение соответствует определенному пути через график, порожденный определенной контекстно-свободной входной грамматикой. Подход, описанный в статье позволяет производить обучение агента за счёт вербальных взаимодействий и даёт возможность абстрагироваться от составления точной последовательности действий в качестве плана, высказывая только пожелания более высокого уровня абстракции. В статье не описана возможность многоагентных коммуникаций с целью распределения ролей агентов в коллективе.

#### 4. Гибридный подход

Как было сказано в п. 1 ни когнитивный ни численный подходы к планированию сами по себе не способны решать многоагентную гетерогенную задачу планирования с использованием в качестве агентов как интеллектуальный роботизированных платформ, так и людей, и одновременно быть открытыми для увеличения участников коллаборации за счёт агентов окружающей среды. Численные методы, позволяющие увеличивать коллаборацию для использования новых искусственных агентов не могут обеспечить полноценную взаимосвязь с человеком, а когнитивным методам не хватает возможностей для создания единого представления знаний среди всех агентов, которое позволило бы с большой скоростью включать новых агентов в процесс планирования. Для таких целей используются гибридные подходы, основанные на психологически правдоподобных способах представления знаний.

Основываясь на психологическом подходе к представлению знаний, которое может быть использовано при осуществлении процесса планирования деятельности, описанном в статьях [Осипов, 2014; Осипов, 2015; Панов, 2015; Панов, 2016] была создана когнитивная архитектура, позволяющая не только выстраивать базис планирования, но и сохранять прецеденты деятельности, облегчая и ускоряя процесс действия в текущих ограничениях для интеллектуальных агентов. В работах описывается знаковый подход к представлению знаний, в котором для всех объектов и агентов окружающей среды, а также для всех действий, отношений на множестве объектов, абстрактных ролей объектов в действиях в картине мира субъекта деятельности имеются психологически правдоподобные четырёхкомпонентные структуры, названные знаками. Знак имеет имя, значение, образ и личностный смысл. Все компоненты знака кроме имени представляются фрагментами семантических сети, состоящих из каузальных матриц, включающих в себя ссылки на другие знаки. Все знаки включены в единую семиотическую сеть, которая

является моделью картины мира интеллектуального агента. В статье [Kiselev, 2017] был предложен многоагентный подход к планированию MultiMap, использующий знаки «Я» и «Другой» для осуществления процесса распределения действий между агентами.

## 5. Коммуникативные действия агентов

В рамках многоагентного подхода к решению задачи планирования в знаковой картине мира предлагается использовать действия по отправке сообщений, позволяющие гетерогенным агентам совместно строить план и связываться друг с другом с помощью сообщений, составленных на некотором подмножестве естественного языка. Для реализации коммуникационных действий между агентами в алгоритме используются речевые акты, основанные на лексемах, интерпретирующих элементы каузальной сети значений интеллектуального агента в словоформы естественного языка. Сообщения агентов составляются с помощью общей для всех агентов каузальной сети значений и могут быть использованы для реализации общения между искусственным агентом и человеком. Требования к искусственным агентам заключаются в наличии средств для реализации кодирования требуемой последовательности действий в естественный язык из личной знаковой картины мира, распознавания и означивания полученных ответов. Каждым из агентов используется знак «Согласовать» для обозначения процесса сложного действия по установлению соединения и совершению коммуникационного акта с другими агентами. Действие «Согласовать» описывает набор поведений агента, каждое из которых применяется в зависимости от требований текущей ситуации. В коммуникационное поведение агента входят действия по объявлению намерений агента через широкополосный канал связи, либо непосредственная отправка сообщения другому агенту лично. Далее представлен алгоритм MultiMAP, в котором описан знаковый подход к многоагентному планированию с использованием сообщений.

Алгоритм 1. Алгоритм MultiMAP

1. manager = Manager\_init(task\_name, domain\_name)
2. agents = manager.start\_agents
3. for agent in agents:
4.     task = agent.ground(manager.task)
5.     agent.plan = agent.map\_search(task)
6.     plans.append(agent.plan)
7. def map\_search(task):
8.     plans = map\_iteration(plans)
9.     longest\_cooperations = find\_longest(plans)



```

10.     min_agents = find_min(longest_cooperations)
11.     with_me = with_me(min_agents)
12.     return with_me
13. for agent in agents:
14.     message = ""
15.     for action, sit in agent.plan:
16.         message += Tmessage(action, sit)
17.     agent.mes = message
18. for agent in agents:
19. if len(set(plans)) > 1:
20.     agent.auction_cooperate(agent.mes)
21. else:
22.     agent.send_message(agent.mes)

```

В алгоритме MultiMAP на шагах 1-2 происходит процесс запуска агентов планирования и на вход алгоритма подается описание задачи на языке ML-PDDL [Alfonso, 2005]. На шагах 3-12 происходит процесс означивания и поиска всех возможных планов агентами, процесс подробно описан в статье [Kiselev, 2017]. Отличия составляют шаги 9-11, на шаге 9 среди всех найденных планов выбираются планы, в которых максимально количество последовательных действий агента, на шаге 10 из полученных планов выбираются планы с наименьшим количеством участвующих агентов и на шаге 11 из оставшихся планов выбирается план с участием планирующего агента. На шагах 13-17 каждый из агентов трансформирует полученный план с помощью лексем естественного языка в сообщении. Алгоритм составления сообщения на естественном языке заключается в сопоставлении знаков, входящих в каузальные матрицы знаков ситуаций на сети значений, включенных в план и шаблонизированных лингвистических предикатов. На шагах 19-22 осуществляется согласование полученных планов. Если планы агентов не совпадают, то осуществляется аукцион по итогам которого выбирается план, за который проголосовало большинство агентов, если же планы агентов совпадают, то все агенты уведомляются об окончании процесса планирования.

Единый способ представления знаний, построенный, основываясь на психологических подходах к деятельности человека, предоставляет возможность использовать подход к многоагентному планированию для коллективов агентов, включающих как искусственные агенты, так и человека. В отличие от других гибридных подходов к многоагентному планированию, в подходе, который описывается в статье, используется единая сеть значений, что позволяет значительно ускорять распознавание команд искусственным агентом и реализацию последующей совместной деятельности.

## Заключение

В статье были описаны численные, когнитивные и гибридные подходы к многоагентному планированию, рассмотрены их недостатки и достоинства. Основываясь на психологически подобной модели представления знаний была сформулирована модель нового гибридного подхода к многоагентному планированию, с использованием коммуникационных действий между агентами на подмножестве естественного языка.

## Список литературы

- [Alfonso, 2005] Alfonso E Gerevini and Derek Long. Plan Constraints and Preferences in PDDL3. Technical Report, pages 1–12, 2005.
- [Carter, 2003] J. Carter and A. A. Ghorbani. 2003. Value centric trust in multiagent systems. In Proceedings of IEEE/WIC International Conference on Web Intelligence (WI 2003).
- [Carter, 2002] J. Carter, E. Bitting, and A. A. Ghorbani. 2002. Reputation formalization for an information sharing multi-agent system. In Computational Intelligence 18, 2, 515–534.
- [Castelfranchi, 2001] C. Castelfranchi and R. Falcone. 2001. Social trust: A cognitive approach. In Trust and Deception in Virtual Societies, pp. 55–90. Kluwer Academic Publishers.
- [FIPA] The Foundation for Intelligent Physical Agents. (2002). *FIPA Communicative Act Library Specification*. *FIPA TC Communication*. Retrieved from <http://www.fipa.org/specs/fipa00037/>
- [Guo, 2016] Guo, M., & Dimarogonas, D. V. (2016). Task and Motion Coordination for Heterogeneous Multiagent Systems With Loosely Coupled Local Tasks, 1–12. <https://doi.org/10.1109/TASE.2016.2628389>
- [Hadad, 2013] Hadad M. [и др.]. Group planning with time constraints / M. Hadad, S. Kraus, I. Ben-Arroyo Hartman, A. Rosenfeld, 2013, 243–291 с.
- [Jakubův, 2015] Jakubův, J., Tožička, J., & Komenda, A. (2015). Multiagent Planning by Plan Set Intersection and Plan Verification. *Proceedings of 7th ICAART*, 173–182. <https://doi.org/10.5220/0005222101730182>
- [Jones, 2006] Jones E.G. [и др.]. Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks 2006. 570–575 с.
- [Kiselev, 2017] Kiselev G.A., Panov A.I. Synthesis of the Behavior Plan for Group of Robots with Sign Based World Model.
- [Matt, 2010] P. A. Matt, M. Morge, and F. Toni. 2010. Combining statistics and arguments to compute trust. In Proceedings of the 9th International Conference on Autonomous Agents and MultiAgent Systems, pp. 209–216, Toronto, Canada.
- [Neville, 2004] B. Neville and J. Pitt. 2004. A computational framework for social agents in agent mediated e-commerce. In Proceedings of the Seventh International Workshop on Trust in Agent Societies, R. Falcone, S. Barber, L. Korba, and M. Singh, Eds., New York, July 2004, 83–91.

- [Parsons, 2011] S. Parsons, Y. Tang, E. Sklar, P. McBurney, and Cai. K. 2011. Argumentation-based reasoning in agents with varying degrees of trust. In Proceedings of the 10th International Conference on Autonomous Agents and MultiAgent Systems. 879–886.
- [Roy, 2005] Roy D. Semiotic schemas: A framework for grounding language in action and perception // *Artificial Intelligence*. 2005. № 1–2 (167). С. 170–205.
- [Sabater, 2005] Sabater, J., & Sierra, C. (2005). Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1), 33–60. <https://doi.org/10.1007/s10462-004-0041-5>
- [Schrodt, 2017] Schrodt F. [и др.]. Mario Becomes Cognitive // *Topics in Cognitive Science*. 2017. С. 1–31.
- [SPHINX-4] Lamere, P., Kwok, P., Gouvea, E., Raj, B., Singh, R., Walker, W., ... Wolf, P. (2003). The CMU SPHINX-4 speech recognition system. *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong, 1, 2–5*.
- [Zhang, 2007] J. Zhang, A. A.Ghorbani, and R. Cohen. 2007. A familiarity-based trustmodel for effective selection of sellers in multiagent e-commerce systems. *International Journal of Information Security* 6, 5, 333–344.
- [Осипов, 2014] Осипов Г.С., Панов А.И., Чудова Н.В. Управление поведением как функция сознания. I. Картина мира и целеполагание // *Известия Российской академии наук. Теория и системы управления*. 2014. № 4. С. 49–62.
- [Осипов, 2015] Осипов Г.С., Панов А.И., Чудова Н.В. Управление поведением как функция сознания. II. Синтез плана поведения // *Известия Российской академии наук. Теория и системы управления*. 2015. № 6. С. 47–61.
- [Панов, 2015] Панов А.И. Представление знаний автономных агентов, планирующих согласованные перемещения // *Робототехника и техническая кибернетика*. 2015. № 4(9). С. 34–40.
- [Панов, 2016] Панов А.И., Яковлев К.С. Взаимодействие стратегического и тактического планирования поведения коалиций агентов в динамической среде // *Искусственный интеллект и принятие решений*. 2016. № 4. С. 68–78.
- [Тарасов, 2002] Тарасов В.Б. От многоагентных систем к интеллектуальным организациям. М.: Эдиториал УРСС, 2002. 352 с.

УДК 004.021

## КОММУНИКАЦИОННЫЙ АСПЕКТ ЗАДАЧИ ИССЛЕДОВАНИЯ ОБЛАСТИ ГРУППОЙ РОБОТОВ С ЛОКАЛЬНЫМ ВЗАИМОДЕЙСТВИЕМ

В.В. Воробьев (*gatus86@mail.ru*)  
НИЦ “Курчатовский институт”, Москва

**Аннотация.** В работе рассматривается коммуникационный аспект задачи группового исследования заданной области. Этот аспект заключается в том, как организовано взаимодействие и обмен данными между роботами в группе. Предлагается механизм взаимодействия, основанный на обмене сигналами, которые интерпретируются роботами, которые их зафиксировали, особым образом так, как будто область, откуда был принят сигнал, уже исследована. Показаны результаты моделирования, подтверждающие принципиальную возможность использования такого механизма для решения данной задачи.<sup>1</sup>

**Ключевые слова:** групповая робототехника, групповое картографирование, локальное взаимодействие.

### Введение

В работе рассматривается коммуникационный аспект задачи совместного исследования заданной области коллективом роботов. Вообще говоря, данный аспект является во многом концептуальным при разработке групповых робототехнических систем. Например, от него зависит степень информированности членов группы об общих целях, задачах и т.д. [Каляев и др., 2009]. Таким образом, характер взаимодействия влияет как на специфику решаемых группой задач, так и на методику их решения. Очевидно, что чем более информирована группа, тем более сложные задачи она может решать.

Если говорить о задаче исследования области, то можно сказать, что она относится к фундаментальным задачам мобильной робототехники, и обычно ее решение сводится к определению дальнейших действий робота, с целью минимизации времени исследования [Burgard et al., 2000]. В случае

---

<sup>1</sup> Работа выполнена при частичной финансовой поддержке грантов РФФИ проект (№ 16-11-00018) и РФФИ (проект № 15-07-07483).

использования группы роботов, коммуникационный аспект также начинает играть важную роль.

Например, если свести исследование области к задаче распределения роботов на территории, где каждый робот исследует только свою часть [Rekleitis et al., 1998], то взаимодействия между роботами, непосредственно во время исследования, может и не быть вовсе. В таком случае важно иметь алгоритм, который оптимально распределит роботов по территории, и сущность, которая запустит его. Однако такую группу практически невозможно адаптировать к изменениям в процессе исследования, например, выбыванию части роботов из строя. Для этого необходима постоянная связь членов группы с сущностью-лидером, которая отслеживала бы изменения обстановки и могла внести коррективы в распределение роботов по территории. Например, в [Churavy et al., 2008] построена централизованная система управления группой роботов, где связь между базовой станцией и роботами осуществляется по радиоканалу. Однако эффективность такого подхода может резко упасть при выбывании лидера или в случае, когда каналы связи между ним и остальными роботами сильно зашумлены, например, в случае сознательного подавления этих каналов противником.

В таком случае возможно более эффективным подходом является подход, где члены группы могут обмениваться информацией друг с другом, а группа не имеет лидера, или, по крайней мере, его функции при решении задачи исследования ничем не отличаются от функций остальных роботов. При этом он может первоначально распределить роботов по территории, но отдельному роботу здесь не назначается какой-то конкретный участок области для исследования, а адаптация группы происходит посредством взаимодействия друг с другом.

Архитектурно можно выделить системы, где каждый робот имеет возможность установить связь со всеми роботами группы, и системы, где есть ограничения на число таких связей. Если рассматривать каждого робота как узел некоей компьютерной сети, то примером первого варианта систем, могут служить сети с полносвязной топологией, а примером второго варианта – сети с неполносвязной. Частным случаем системы с ограничениями являются системы, использующие локальное взаимодействие, т.е. те системы, где роботы могут взаимодействовать исключительно со своими ближайшими соседями [Карпов, 2013].

Безусловно, если не рассматривать техническую сторону решения, то предпочтительнее, с точки зрения обеспечения более высокой степени информированности коллектива, полносвязная топология. Однако технически это труднореализуемо: при числе роботов в коллективе порядка десятков единиц существенной проблемой становится установка и поддержание каналов связи между всеми роботами. Например, в случае

использования Bluetooth или ZigBee необходима установка большого числа, сравнимого с числом роботов, устройств на каждого робота, что конструктивно труднореализуемо. Не стоит забывать и о том, что множество работающих устройств перегружают эфир. Кроме того, речь идет о группе роботов, соответственно они могут двигаться, выполняя свою задачу. Следовательно, топология может меняться из-за того, что ряд роботов оказалась вне границ “слышимости”, а ряд роботов, наоборот, добавился в общую сеть. В таком случае одним из вариантов организации взаимодействия может быть вариант с использованием самоорганизующихся сетей (ad-hoc) [Вишнеvский и др., 2005].

Например, в [Couceiro et al., 2014] рассматривается решение задачи исследования местности группой роботов, в основе которого лежат мобильные ad-hoc сети (mobile ad-hoc network MANET). Суть решения заключается в том, что есть роботы-исследователи (scout) и роботы-наблюдатели (ranger). Первые занимаются непосредственно исследованием местности, а вторые обеспечивают связь, контролируя движение первых. Однако даже в этом случае авторы указывают на сложность реализации такого рода систем из-за их динамической нестабильности.

Еще одна система подобного рода представлена в [Burgard et al., 2005], где решается, в том числе и задача картографирования. Роботы образуют кластеры, внутри которого они могут обмениваться информацией с любым членом этого кластера, координировать совместные действия. Кластеры образуются, когда роботы могут “слышать” друг друга. В вырожденном случае, когда никто друг друга не видит, они действуют индивидуально.

Если рассматривать системы с ограничениями на количество связей, то недостатки систем с полносвязной топологией здесь не так явны, однако достигнуть такой же степени информированности каждого члена коллектива если и возможно, то это занимает больше времени и/или требует дополнительных программно-аппаратных решений.

В качестве примера можно выделить работу [Zhu et al., 2013], где рассматривается задача исследования Марса коллективом роботов. Работа, в основном, посвящена разработке навигационной системы для этого коллектива. Однако указывается, что для итеративной процедуры уточнения координат роботы связываются с соседями, используя самоорганизующийся режим передачи цифровой информации с временным разделением канала.

В [Воробьев, 2015] рассматривается задача коллективной рекогносцировки местности, где коллектив роботов использует только локальное взаимодействие роботов. Технически, система взаимодействия – ИК-канал связи с реализацией протокола передачи данных RC-5 [Карпова,

2016]. С помощью нее робот может “общаться” со своими ближайшими соседями.

Несмотря на технические трудности, реализации систем с полностью связанной топологией существуют, в том числе в виде реальных программно-аппаратных комплексов, а сам подход достаточно успешно развивается. Если говорить о задаче группового исследования местности, можно сказать, что это, во многом, обусловлено подходом к решению задачи – роботы рассматриваются как узлы вычислительной сети, топология которой может динамически меняться. Это имеет свои преимущества – обеспечивается достаточно интенсивный обмен данными между роботами, что важно, например, в задаче коллективного SLAM (см. например [León et al., 2009]), и позволяет коллективу одновременно создавать общую карту местности, обмениваясь локальными картами друг с другом непосредственно в процессе исследования. Однако этот подход достаточно ресурсоемок, но самое главное – коллектив роботов не является компьютерной сетью в полном смысле этого слова [Карпов, 2013], а их взаимодействие не сводится только лишь к обмену информацией. При этом остается проблема зашумленных каналов связи, по которым необходимо передавать большие объемы данных, указанная выше.

В работе рассматривается возможность использования неявной коммуникации в коллективе роботов, решающих задачу исследования местности, где взаимодействие между ними носит локальный характер – робот может взаимодействовать только со своими соседями. Выбор системы, использующей локальное взаимодействие, обусловлен не только тем, что роботы будут технически проще в реализации, но и тем фактом, что в реальной жизни при общении невозможно передать информацию всему коллективу при его численности более определенного порога. Если рассматривать муравьев как объект подражания групповой робототехники [Карпов, 2016], становится очевидно, что связь “все-со-всеми” просто невозможна.

Другой особенностью является неявный характер коммуникации – роботы не обмениваются данными в процессе исследования. Каждый робот постоянно генерирует специальный сигнал, и когда его “слышат” другие роботы, они интерпретируют область, откуда сигнал получен как уже исследованную. В отличие от похожего метода потенциалов, пример которого представлен, например в [Karpov et al., 2016], интерпретированная единожды как уже исследованная область остается таковой до конца исследования, т.е. потенциал не зависит от времени. Кроме того, упрощен механизм трактовки полученного сигнала, что позволяет решать задачу группового исследования в условиях высокого уровня помех или сознательного подавления каналов связи противником.

## 1. Постановка задачи

Рассмотрим следующую задачу. Существует  $N$  роботов, взаимодействие которых друг с другом носит локальный характер. Перед ними стоит задача исследования некоторой заданной местности.

Каждый робот имеет область видимости, радиус которой равен  $R$ . Область, в которой робот может заметить другого робота совпадает с областью видимости  $R$ , при этом можно определить направление, откуда был получен сигнал. Всего есть 8 направлений. Аппаратная реализация системы локальной связи, позволяющей определить направление, откуда был получен сигнал описано в [Карпова, 2016] и представляет собой инфракрасный приемо-передатчик, где каждый приемник экранирован так, чтобы обеспечить получение сигнала только из своего квадранта. Схематично, определение направления выглядит следующим образом (Рис. 2): робот слева получает на правый приемник сигнал от робота справа. Стоит отметить, что в нормальных условиях в таком случае, и робот справа принимает сигнал на левый датчик от робота слева.

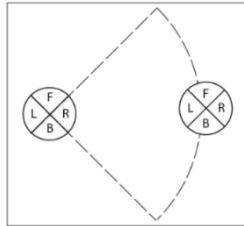


Рис. 2. Определение направления, откуда был получен сигнал.

За препятствиями видимость отсутствует. Исследуемая местность поделена на квадраты. Двигаться роботы могут в любой из 4х смежных с текущем квадраты. Исследование области происходит следующим образом: робот выбирает ближайший неисследованный квадрат, строит к нему путь по алгоритму  $A^*$  и начинает движение по маршруту. Как только квадрат попадает в поле его видимости, он считается исследованным, и робот выбирает следующий неисследованный квадрат.

В отличие от [Воробьев, 2015] и [Karpov et al., 2016], где передавались координаты исследованной области (Рис. 3), здесь передается только направление, откуда был получен сигнал (Рис. 4), и это направление считается уже исследованным. Стоит отметить тот факт, что при встрече роботов друг с другом, полученный сигнал можно интерпретировать как “Не ходи сюда”, при этом он не содержит никакого сообщения –



достаточно просто наличие соответствующего сигнала. Это позволит роботам избегать уже исследованных их соседями областей.

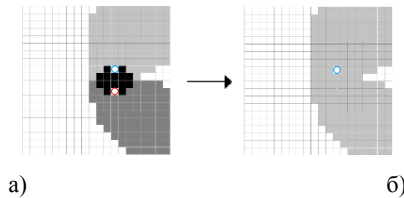


Рис. 3. Непосредственный обмен областями. а) - до обмена, б) – после, для верхнего робота. Черным отмечена область, которую видят оба робота

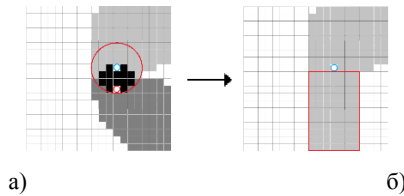


Рис. 4. Обмен сигналами. а) - до обмена, б) – после, для верхнего робота. Черным отмечена область, которую видят оба робота

При получении сигнала робот отмечает область, откуда он пришел так, как будто он был там сам, что позволит уменьшить время исследования всей области коллективом. Отмечаемая область имеет форму прямоугольника (Рис. 4,б), короткая сторона которого имеет длину, равную диаметру обзора робота (Рис. 4,а, отмечен красным кругом), а длинная сторона доходит от робота до края исследуемой области. При этом алгоритм действий робота, который получил сигнал можно описать следующим образом:

Алгоритм 1. Действия при получении сигнала.

- |   |
|---|
| <ol style="list-style-type: none"> <li>1. Определить, с какой стороны получен сигнал.</li> <li>2. Пометить квадраты с этой стороны как уже исследованные (см. Рис. 4).</li> </ol> |
|---|

## 2. Вычислительные эксперименты

Среда моделирования, где проводились вычислительные эксперименты, представляет собой написанную на языке C++ программу, состоящую из графического и логического модулей. Выбор в пользу языка и самостоятельной разработки среды моделирования обусловлен

необходимостью быстрого переноса разработанного механизма на реальную группировку роботов.

Вычислительные эксперименты проводились для двух моделей полигона: со случайно расставленными препятствиями (Рис. 5а) и с моделью лабораторного полигона (Рис. 5б). Полигоны поделены на клетки, размером 20x20 см. Размеры полигонов – 20x20 клеток.

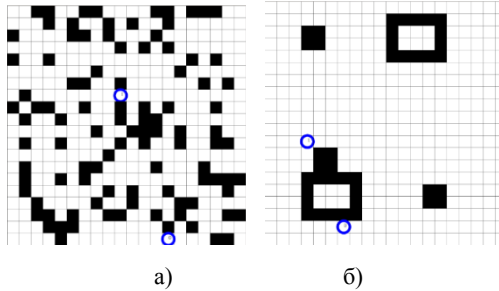


Рис. 5. а – полигон со случайными препятствиями, б – лабораторный полигон.  
Черные квадраты – препятствия, круги – роботы

Роботы расставляются произвольным образом. Для первого случая варьировалось число препятствий: от 5% до 30% с шагом в 5%. Также в обоих случаях варьировалась дальность обзора – от 2 до 4-х клеток. Для каждого набора данных проводилось 100 итераций. Измерялось время, необходимое для того, чтобы у всех роботов все клетки были помеченные как исследованные. Кроме того, оценивалось качество процесса исследования – по количеству квадратов с препятствиями, которые не попали ни разу в зону видимости ни одного робота. Чем выше этот показатель, тем хуже качество.

Для начала рассмотрим эти показатели для полигона с моделью лабораторного полигона. На Рис. 6 показана зависимость среднего числа необнаруженных препятствий в процентах от их общего количества от числа роботов для  $R=2$  и  $R=4$  соответственно.

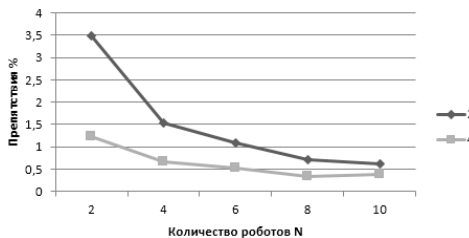


Рис. 6. Средний процент необнаруженных препятствий для лабораторного полигона

Можно заметить, что для меньшего числа роботов ошибка выше. Это объясняется тем, что в таком случае, из-за особенностей механизма трактовки сигнала, чаще встречаются области, которые помечаются как уже исследованные для всех роботов, однако ни один из них их не посетил. На Рис. 6 показана зависимость времени исследования от числа роботов для  $R=2$  и  $R=4$  соответственно.

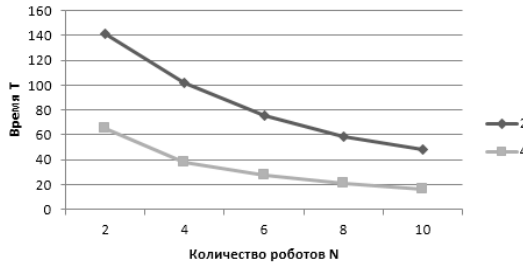


Рис. 7. Время исследования лабораторного полигона

Наблюдается очевидный результат – для большего числа роботов и при большем  $R$  время исследования становится меньше.

Рассмотрим вариант со случайно расставленными препятствиями. Сразу стоит отметить достаточно очевидный аспект – чем меньше число препятствий на полигоне, тем меньше ошибка и меньше время исследования. Это подтверждают и данные вычислительных экспериментов. Вместо этого важно сравнить предлагаемый механизм с другими: непосредственным обменом картами, методом потенциалов и подходом, где роботы вообще не взаимодействуют, и каждый исследует всю территорию. На **Ошибка! Источник ссылки не найден.** представлена зависимость величины ошибки для 10 роботов от процента заполнения территории.

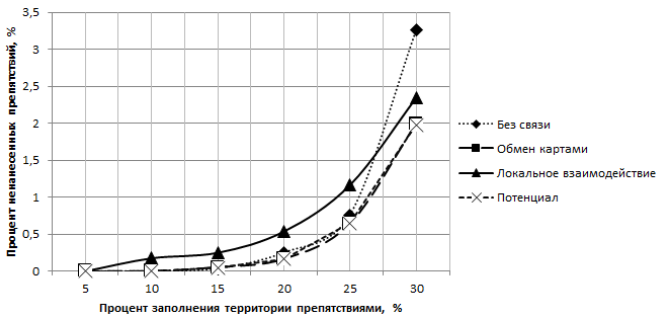


Рис. 8. Средний процент не нанесенных препятствий для разных методов

Можно заметить, что обмен картами и метод потенциалов дают практически идентичные значения ошибки, что обусловлено тем, что, в конечном итоге, хотя бы один из роботов посетит неисследованную область. Предлагаемый механизм показывает результаты хуже, однако, в абсолютных значениях разница с первыми двумя невелика. Однако, если сравнить время исследования (Рис. 9), то видно, что предлагаемый алгоритм самый быстрый из всех.

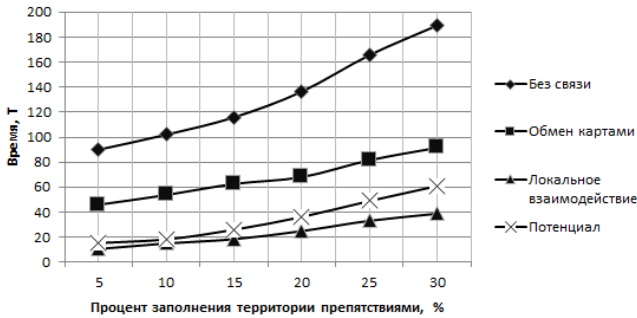


Рис. 9. Время исследования для разных методов.

Это обусловлено тем, что в модели на обмен картами требуется достаточно большое время, что имитирует потерю данных при помехах в канале связи. Метод потенциалов, по сравнению с ним, показывает результаты лучше, однако из-за возникающего перекрытия исследуемых областей роботам требуется больше времени для того, чтобы закончить процедуру.

## Заключение

В работе показана принципиальная возможность использования механизма неявной коммуникации для решения задачи исследования заданной области коллективом роботов, использующих локальное взаимодействие. Характерной особенностью предлагаемого решения является отсутствие необходимости обмена какой-либо информацией между роботами, кроме сигнала, о том, что он находится в радиусе обнаружения другим роботам. Исходя из полученных зависимостей времени и ошибки исследования, можно сделать вывод о том, что такой механизм больше подходит для задач, где важно быстро получить данные в ущерб их точности, например, сбор данных в агрессивной среде или работа под воздействием противника, в местах, где трудно гарантировать стабильный канал связи.

## Список литературы

- [Burgard et al., 2000] Burgard W., Moors M., Fox D., Simmons R., Thrun S. Collaborative Multi-Robot Exploration. // *IEEE International Conference on Robotics and Automation (IRCA)*. pp. 476–481.
- [Burgard et al., 2005] Burgard W., Moors M., Stachniss C., Schneider F. E. Coordinated multi-robot exploration. // *IEEE Transactions on Robotics*, 21(3), pp.376–386.
- [Churavy et al., 2008] Churavy C., Baker M., Mehta S., Pradhan I., Scheidegger N., Shanfelt S., Rarick R., Simon D. Effective implementation of a mapping swarm of robots. , pp.28–33.
- [Couceiro et al., 2014] Couceiro M. S., Figueiredo C. M., Rocha R. P., Ferreira N. M. F. Darwinian swarm exploration under communication constraints : Initial deployment and fault-tolerance assessment. *Robotics and Autonomous Systems*, 62(4), pp.528–544.
- [Karpov, et al., 2016] Karpov V., Migalev A., Moscowsky A., Rovbo M., Vorobiev V. Multi-robot Exploration and Mapping Based on the Subdefinite Models. // In *Interactive Collaborative Robotics, First International Conference, ICR 2016, Budapest, Hungary, August 24-26, 2016, Proceedings*. Springer International Publishing Switzerland, C. 143–152
- [León et al., 2009] León A., Barea R., Bergasa L. M., López E., Ocaña M., Schleicher D. SLAM and map merging. *Journal of Physical Agents*, 3(1), pp.13–23.
- [Rekleitis et al., 1998] Rekleitis, I., Dudek, G. & Milius, E., 1998. Accurate Mapping of an Unknown World and Online Landmark Positioning. In *Proc.of Vision Interface (VI)*. C. 455–461.
- [Zhu et al., 2013] Zhu C., Zhang S., Dammann A., Sand S., Henkel P., Gunther C. Return-to-Base Navigation of Robotic Swarms in Mars Exploration Using DoA Estimation. // *ELMAR, 2013 55th International Symposium*.
- [Вишнеvский и др., 2005] Вишнеvский В.М., Ляхов А.И., Портной С.Л., Шахнович И.В., *Широкополосные беспроводные сети передачи информации*, М. Техносфера, 2005, 592с.
- [Воробьев, 2015] Воробьев В.В. Решение задачи групповой рекогносцировки с использованием локального взаимодействия роботов. // *Беспилотные транспортные средства с элементами искусственного интеллекта*. С. 34–41.
- [Каляев и др. 2009] Каляев И.А., Гайдук А.Р., Капустян С.Г., *Модели и алгоритмы коллективного управления в группах роботов* М: Физматлит, 2009, 292с.
- [Карпов, 2016] Карпов В.Э Модели социального поведения в групповой робототехнике. *Управление большими системами*, (59), С.165–232.
- [Карпов, 2013] Карпов В.Э Управление в статических роях. Постановка задачи.// *VII-я Международная научно-практическая конференция “Интегрированные модели и мягкие вычисления в искусственном интеллекте.”* С. 730–739.
- [Карпова, 2016] Карпова И.П. К вопросу о представлении маршрута для робота в задаче фуражирования. // *Конференция по искусственному интеллекту 2016*. С. 169–178.
- [Карпова, 2016] Карпова И.П. Псевдоаналоговая коммуникация в группе роботов. *Мехатроника, автоматизация, управление*, 17(2), С.94–101.

УДК 62-519, 004.418, 004.716

## УДАЛЁННАЯ КОММУНИКАЦИЯ МНОЖЕСТВА ПОЛЬЗОВАТЕЛЕЙ С МУЛЬТИАГЕНТНОЙ РАСПРЕДЕЛЁННОЙ СИСТЕМОЙ НА ОСНОВЕ МОБИЛЬНЫХ АВТОНОМНЫХ БТС

А.Р. Гамаюнов (*agamayunov@le-talo.ru*)

Е.М. Притоцкий (*epritotskiy@le-talo.ru*)

П.К. Герасимов (*pgerasimov@le-talo.ru*)

Д.А. Егоров (*degorov@le-talo.ru*)

Владимирский государственный университет  
имени А.Г. и Н.Г. Столетовых, Владимир

**Аннотация.** Статья содержит постановку и решение задачи удаленной коммуникации с мультиагентным комплексом на основе автономных мобильных беспилотных транспортных систем (БТС). Основной акцент сделан на беспилотные летательные аппараты (БЛА), как активно развивающиеся на данный момент вид БТС, хотя приведенные принципы справедливы для всех видов БТС. Проведен обзор основных известных способов передачи информации между БЛА и пунктом управления оператора, выявлены проблемы, не позволяющие решить данную задачу, используя описанные технологии. Предложен и апробирован способ решения задачи, создан проект комплекса, демонстрирующего предложенное решение.

**Ключевые слова:** БТС, БЛА, мультиагентная система, протокол коммуникации, удалённая связь, автономная миссия, передача видеопотока.

### Введение

В настоящее время в области робототехники с увеличением количества БТС всё более актуальной является задача их централизованного управления. Эта задача является логическим развитием мультиагентных систем, аналогично области IoT (Internet of Things) [Ashton, 2009], в которой отдельные блоки управления на прошлом этапе развития объединяли в группу внутри локальной домашней сети, а на данный момент активно развивается направление администрирования и управления этими группами из удаленного пункта управления, как

например в IIoT (Industrial Internet of Things). Система сбора показаний бытовых счётчиков, система анализа технического состояния станков, мониторинга внештатных ситуаций и количества материалов на крупном заводе на данный момент во многих случаях имеет доступ через интернет. Но эти технологии невозможно использовать в области робототехники из-за специфики протоколов и технологий, устоявшихся в этой области, что говорит о возникающей проблеме, которую необходимо решать.

## 1. Обзор

К настоящему времени хорошо исследован процесс коммуникации между пунктом управления и БЛА по радиоканалу. Средствами полудуплексной связи в простейшем случае на БЛА передаются широтно-импульсной модуляции (ШИМ) данных в канале управления, которые могут так же шифроваться фазово-импульсной модуляцией (ФИМ). Данный способ управления позволяет решать множество прикладных задач, управляя беспилотником вручную [Шилов, 2014] (рисунок 1.а).

Компания DJI имеет свой проприетарный полнодуплексный протокол передачи Lightbrige, который позволяет передавать не только базовые команды управления, но и телеметрию, сложные команды построения миссии и цифровой видео сигнал HD качества с задержкой не более 0,4 сек. Однако данный протокол не предусматривает механизма автономного выполнения миссии без связи с базовой станцией, которой служит мобильный телефон [Birnbach, 2017] (рисунок 1.б).

Протокол Micro Aero Vehicle Link (MAVLink) также использует полнодуплексную связь. Он описывает протоколы обмена данными, редактирования миссии и параметров БТС, что даёт возможность автономно выполнять заранее записанную миссию (рисунок 1.в). Протокол предусматривает взаимодействие между многими (до 255) БТС Flight Control Unit (FCU) и базовыми станциями Ground Control Unit (GCU) внутри сети, что должно быть достаточно при использовании радиоканала для передачи данных. Данный протокол часто используется совместно с первым вариантом полудуплексной связи при ручном управлении. Поскольку MAVLink также имплементирует базовые команды управления, он может использоваться и для ручного управления, например, через подключённый к базовой станции джойстик или используя виртуальный джойстик на экране базовой станции [Meier, 2011].

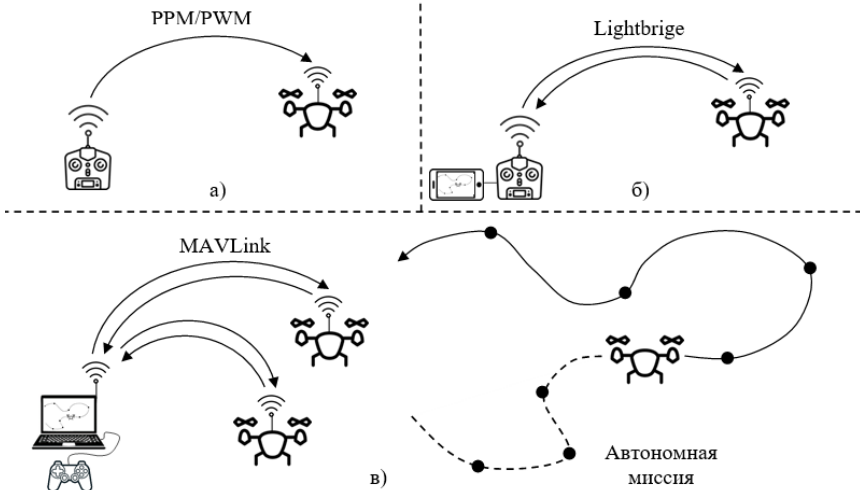


Рис. 1. Способы управления БЛА

Описанные выше варианты коммуникации предполагают передачу данных по радиоканалу, что не позволяет на их основе строить систему управления распределёнными БТС. Стоит отметить, что существует ПО (QGroundControl, MAVproxy, MAVros и др.), позволяющее передавать протокол MAVLink по TCP или UDP, что даёт возможность создавать удалённый канал управления через интернет, если на одной из сторон имеется статический IP адрес, как показано на рисунке 2.а [Alejo, 2015]. При этом для передачи данных используется сеть интернет (либо локальная сеть), доступ к которой базовая станция может получить через GSM сеть, что существенно повышает её мобильность, а клиент может подключиться из любой точки земли. При таком виде коммуникации накладные задержки канала передачи данных на текущем уровне качества мобильной связи не позволяют управлять БТС полностью вручную. Однако данная особенность не критична при редактировании настроек, полуручном управлении через задание точки перемещения, передачей данных и миссии, поскольку протокол MAVLink предусматривает проверку переданных данных на верхнем уровне. Существует возможность создания канала управления, если на клиенте и на базовой станции нет статического адреса, используя промежуточный прокси-сервер с известным статическим адресом, на котором по туннелю Secure Shell (SSH) прокинуты порты одного из участников информационного обмена, как показано на рисунке 2.б.



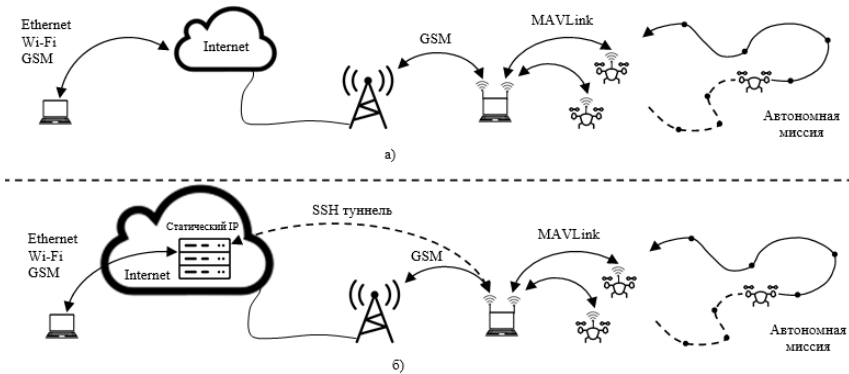


Рис. 2. Способы удалённо управления

Однако описанные выше средства коммуникации не позволяют взаимодействовать одновременно с несколькими удалёнными БТС и не предоставляют доступ множеству клиентов, что не позволяет создавать коммуникацию между распределёнными агентами с разделением доступа. Также проблемой является несовместимость двух самых распространённых протоколов управления БТС: MAVLink и DJI. Для решения этих проблем, необходимо решить следующие задачи:

Задача 1 – описать протокол верхнего уровня, позволяющий унифицировать взаимодействие с агентами, использующими наиболее распространённые протокол управления - MAVLink или DJI,

Задача 2 – предложить способ доступа к БТС множеству клиентов и механизм разделения доступа между ними,

Задача 3 – предложить способ расширения набора данных передаваемых клиенту.

Решая данные задачи, можно получить универсальный инструмент для одновременного управления большим количеством территориально распределённых БТС с целью решения глобальных задач по автоматическому планированию траекторий для навигации БТС [Яковлев, 2015] и совершенствованию алгоритмов, позволяющих организовать роевое взаимодействие в группе агентов [Varela, 2003].

## 2. Универсальный протокол

Не смотря на различия протоколов DJI и MAVlink, сами БЛА имеют практически идентичные примитивы управления и общую функциональность. Они являются достаточно «гибкими», поскольку

рассчитаны на БЛА различных моделей, что проявляется в множестве опциональных параметров вызова команд управления. Обмен обязательными сообщениями такими как heartbeat в MAVLink и служебными сообщениями в dji обеспечивают посредники - MAVros и DJI SDK, к которым имеется доступ через ROS API и DJI API соответственно. Эти интерфейсы можно использовать для получения доступа к телеметрии и управлению БЛА, но они не только платформозависимые, но и перенимают ограничения протоколов нижнего уровня. Например, при создании миссии, в DJI запрещено указывать точку перемещения ближе, чем 1 метр от предыдущей точки, а в MAVLink действия при выполнении миссии, такие как «изменить угол наклона камеры» или «сделать фото», считаются отдельными точками, которые можно указать с одинаковыми координатами.

Реализация протоколов отличается и на аппаратном уровне. Для получения доступа к API ROS необходим одноплатный компьютер с ОС LINUX – так называемый компьютер «компаньон», подключаемый к полётному контроллеру через Universal Asynchronous Receiver-Transmitter (UART) и находящийся на борту БЛА. DJI API рассчитан на использование на мобильном телефоне Android, подключаемому к передатчику Lightbridge, которым обычно служит пульт ручного управления дроном, однако вместо мобильного телефона или планшета можно использовать одноплатный компьютер «компаньон» с ОС Android.

Для решения задачи создания универсального протокола необходимо создать программы-трансляторы из описываемого протокола в вызовы DJI API и ROS API. Для DJI API это будет программа на языке Java, для ROS API на языке Python или C++. Назовём данные программы Icommander1 и Icommander2 соответственно, как показано на рисунке 3.

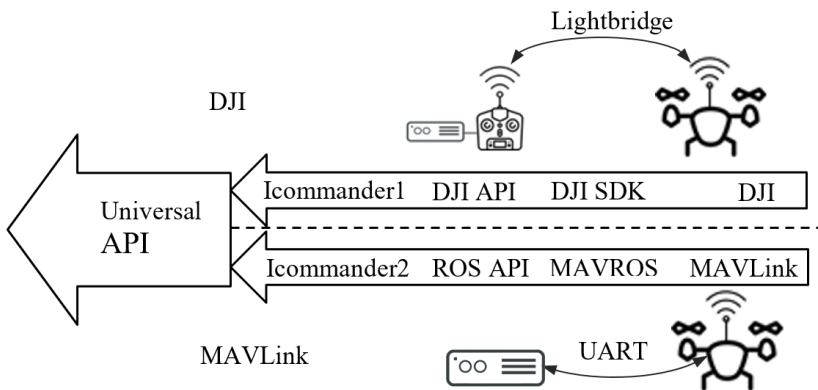


Рис. 3. Формирования универсального интерфейса

Поскольку в программе-трансляторе используется описание бизнес-логики и никаких емких вычислений, то для ROS API рациональнее использование более высокоуровневый язык программирования Python. Эти программы реализуют API с единой сигнатурой вызова функций, что позволяет реализовывать алгоритмы управления БТС независимо от различий моделей беспилотников. Краткое описание протокола представлено в таблице 1. Сообщения передаются по TCP-сокету, что позволяет создавать постоянное соединение в рамках одной сессии и даёт определённую гарантию доставки данных.

Таблица 1

## Описание протокола

Сообщение	Содержание	Описание	Частота отправки
drone-connect	drone_id, name, video	Служит для идентификации БТС и получения ссылки на видео поток.	Однократно при подключении Icommander к клиенту.
drone-telemetry	latitude, longitude, altitude, direction, rssi	Данные телеметрии о глобальных координатах, направлении и уровне сигнала БТС	Не менее 1 сообщения в сек. от Icommander к клиенту
drone-battery	percentage, current, voltage, cells	Данные о напряжении, потребляемом токе и количестве ячеек аккумулятора БТС	Не менее 1 сообщения в сек. от Icommander к клиенту
drone-state	Armed, mode	Данные о текущем режиме работы БТС	Не менее 1 сообщения в сек. от Icommander к клиенту
mission-push	Settings [speed, yaw_strategy, and finish], points [latitude, longitude, altitude, current, actions [wait, pitch, rotate, take_photos, start_video, stop_video...]....] ]	Миссия для автономного выполнения БТС	Однократно от клиента к Icommander
mission-status	Settings [speed, yaw_strategy, and finish], points [latitude, longitude, altitude, current, actions [wait, pitch, rotate, take_photos, start_video, stop_video...]....] ]	Копия миссии для подтверждения получения от клиента и контроля выполнения	При изменении статуса миссии

remote_stick	x,y,z,yaw	Сообщение для ручного управления БЛА при малых задержках сети	При изменении положения органов управления на клиенте
go_to_point	latitude, longitude, altitude	Альтернативный способ ручного управления при больших задержках сети	При указании координат точки клиентом

### 3. Способ множественного доступа

Описанный выше протокол предоставляет универсальный интерфейс управления группой БТС одним клиентом, с которым идёт обмен сообщениями. Для предоставления доступа управления многим пользователям коммуникация должна вестись через посредника, которым выступает виртуальный сервер, в задачи которого так же входит задача разделения доступа. Сервер по сути является одним клиентом, который имеет доступ к управлению всеми подключёнными к нему БТС. Согласно протоколу сервер при подключении БТС получает уникальный идентификационный номер DroneID. Задачу разделения доступа можно решить, введя регистрацию UserID при подключении клиента к серверу управления. Таким образом при хранении в базе данных этих идентификаторов и сопоставлением их между собой можно манипулировать уровнями доступа к управлению БТС разными клиентами. При этом клиенты могут получать доступ к системе управления распределёнными БТС через обычный веб интерфейс, что избавляет от проблемы платформозависимого приложения, как показано на рисунке 4.

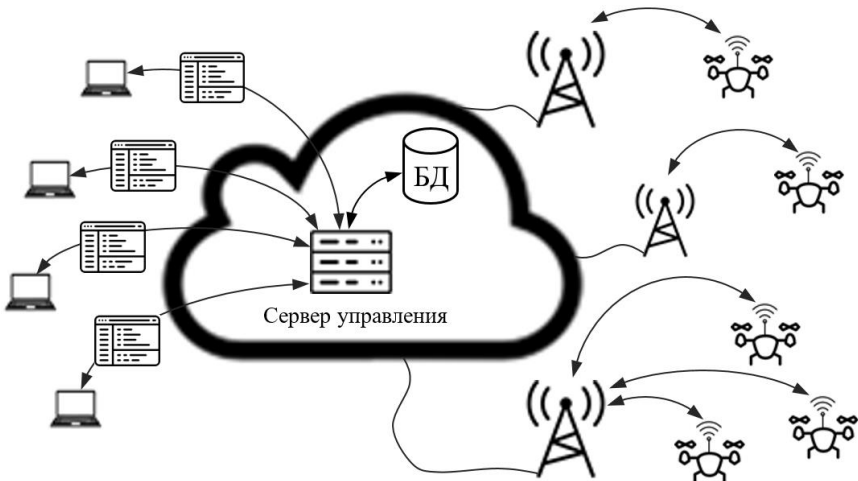


Рис. 4. Архитектура системы управления

#### 4. Передача фото/видео

Предложенный выше протокол предоставляет канал управления и передачу данных телеметрии, однако при этом остаётся не решённым вопрос передачи одного из наиболее важных видов информации - фото/видео материала, позволяющего ориентироваться оператору БТС в пространстве и принимать решения при ручном или полуручном способе управления. Для постобработки полётных данных можно предложить добавить в описанный протокол сообщения, содержащие фотоматериал, однако эти данные будут передаваться с большой задержкой, так как займут большую часть канала передачи данных через сеть GSM, что увеличит задержку передачи данных телеметрии и управления. Для решения данной задачи рациональнее использование одного из способов транслирования сжатого видеопотока через видеосервер. Это позволит снизить требования к каналу передачи данных, поскольку видеопоток в этом случае передаётся по отдельному каналу и в зависимости от качества канала передачи данных может динамически изменять битрейт видеопотока. Способ передачи видеопотока через видеосервер позволит использовать стандартные инструменты управления видеопотоками, такими как запись, хранение, предоставление доступа множеству клиентов, разделение доступа, перекодирование и т.п. Видеопоток при передаче кодируется кодеком h.264 на компьютере компаньоне БТС, упаковывается в контейнер AVI, и передаётся по протоколу Real Time Messaging Protocol (RTMP) на видеосервер, как показано на рисунке 5. Далее сервер управления БТС получает ссылку на видеопоток, которую предоставляет авторизированным клиентам. При наличии функции Digital Video Recorder (DVR) на сервере, как у Simple RTMP Server (SRS), видеосервер может параллельно вести запись видеопотока на жесткий диск.

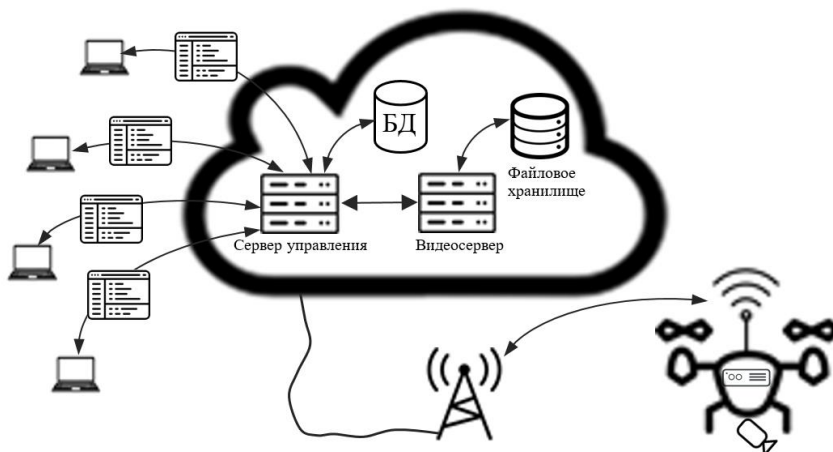


Рис. 5. Трансляция и запись видеопотока

## Заключение

Несмотря на то, что в данной статье для примера были использованы БТС на основе беспилотных летательных аппаратов, предложенный способ коммуникации применим ко всем видам БТС.

В связи с активным развитием теории группового взаимодействия БТС, предложенный способ коммуникации может быть использован исследователями как инструмент для реализации теоретических разработок группового управления, в том числе при групповом управлении БТС с элементами искусственного интеллекта, как виртуальными БТС в режиме Software in the Loop (SITL), так и настоящими.

Предложенный способ удалённого управления был реализован в комплексе авторами статьи и опробован для нужд Главного управления МЧС по Владимирской области, где показал возможность применения описанного инструмента в практических целях. В ходе испытаний был получен устойчивый удаленный канал связи оператора с БЛА на расстоянии более 10 км. Оператор задавал маршруты для автономного выполнения миссии, управлял БЛА в ручном режиме через интернет и произвел разведку местности на наличие очагов возгорания. Разработанный комплекс для автономного мониторинга местности был представлен в рамках X Международного салона средств обеспечения безопасности «Комплексная безопасность – 2017» [МЧС, 2017].

## Список литературы

- [Ashton, 2009] Ashton K. That ‘Internet of Things’ Thing. In the real world, things matter more than ideas // RFID Journal, 2009.
- [Шилов, 2014] Шилов К. Е. Разработка системы автоматического управления беспилотным летательным аппаратом мультироторного типа // ТРУДЫ МФТИ, 2014. — Том 6, № 4. – С. 139-152.
- [Birnbach, 2017] Birnbach S., Baker R., Martinovic I. Wi-Fly?: Detecting Privacy Invasion Attacks by Consumer Drones. NDSS 2017: Network and Distributed System Security Symposium.
- [Meier, 2011] Meier L., Tanskanen P., Fraundorfer F., Pollefeys M. The PIXHAWK open-source computer vision framework for MAVs // International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXVIII-1/C22, 2011 ISPRS Zurich 2011 Workshop, 14-16 September 2011, Zurich, Switzerland.
- [Alejo, 2015] Alejo D., del Arco J. C., Arrue B. C., Cobano J.A., Heredia G., Ollero A. Multi-UAV ground control station for gliding aircraft // 23rd Mediterranean Conference on Control and Automation (MED), 2015, Torremolinos, Spain.
- [Яковлев, 2015] Яковлев К. С., Баскин Е. С., Андрейчук А. А. Метод автоматического планирования совокупности траекторий для навигации беспилотных транспортных средств // Управление большими системами, 2015, Выпуск 58, С. 306-342.
- [Varela, 2003] Varela G., Saamaño P., Orjales F., Deibe A., López-Peña F., Duro R. J. Swarm Intelligence based Approach for Real Time UAV Integrated // IEEE, Group for Engineering Research University of A Coruña Ferrol, 2003, Spain.
- [МЧС, 2017] ГУ МЧС по Владимирской области, Пресс-центр. - <http://33.mchs.gov.ru/pressroom/news/item/5406387>

**Авторский указатель**

Андрейчук А.А.	107	Кулинич А.А.	193
Бодунков Н.Е.	158	Лавренов Р.О.	76, 96
Боковой А.В.	36	Леонард А.В.	148
Брискин Е.С.	148	Магид Е.А.	65,76,87,96
Бритков В.Б.	175	Михайлов В.А.	25
Буйвал А.К.	14	Михайлов Н.А.	126
Воробьев В.В.	216	Московский А.Д.	56
Ву Д.К.	183	Нгуен В.В.	183
Гавриленков М.А.	14	Павловский В.Е.	175
Гамаюнов А.Р.	226	Панов А.И.	204
Герасимов П.К.	226	Пилипенко О.Г.	25
Герасимов Ю.А.	87	Притоцкий Е.М.	226
Горбов И.А.	148	Ройзензон Г.В.	175
Добрынин Д.А.	118	Ронжин А.Л.	183
Егоров Д.А.	226	Сагитов А.Г.	65, 87
Жидков В.Н.	165	Соленая О.Я.	183
Закиев А.А.	76	Сорокоумов П.С.	46
Зенкевич С.Л.	136	Цзяньвень Хо	136
Карпов В.Э.	175	Чжу Хуа	136
Ким Н.В.	158, 165	Шабалина К.С.	65
Киселёв Г.А.	204	Яковлев К.С.	107



*Для заметок*

---

*Для заметок*

---

**Российская ассоциация искусственного интеллекта  
Высшая школа информационных технологий и  
информационных систем  
Казанского федерального университета**

---



**БТС  
ИИ** 2017

**ЧЕТВЕРТЫЙ ВСЕРОССИЙСКИЙ  
НАУЧНО-ПРАКТИЧЕСКИЙ СЕМИНАР  
«БЕСПИЛОТНЫЕ ТРАНСПОРТНЫЕ  
СРЕДСТВА С ЭЛЕМЕНТАМИ  
ИСКУССТВЕННОГО ИНТЕЛЛЕКТА»**

5-6 октября 2017  
г. Казань, Республика Татарстан, Россия

**Труды семинара**

ISBN 978-5-93962-833-4



Подписано в печать 05.09.2017. Формат 60x84  $\frac{1}{16}$ .  
Бумага офсетная. Печать офсетная.  
Гарнитура «Times». Усл. печ. л. 13,95.  
Тираж 200 экз. Заказ 09-17/13-1

Отпечатано с готового оригинал-макета



420108, г. Казань, ул. Портовая, 25а.  
Тел./факс: (843) 231-05-46, 231-08-71, 231-04-19.  
E-mail: [citlogos@mail.ru](mailto:citlogos@mail.ru)  
[www.logos-press.ru](http://www.logos-press.ru)