

УДК 681.5+004.8

## К ВОПРОСУ ОБ ОРИЕНТАЦИИ РОБОТОВ НА ОСНОВЕ ВИЗУАЛЬНЫХ ОРИЕНТИРОВ И КОМПАСА

И.П. Карпова (*karpova\_ip@mail.ru*)  
НИУ ВШЭ, Москва

В работе рассматривается метод использования визуальных ориентиров для запоминания пройденного пути мобильным роботом (аниматом), основанный на механизме навигации муравьев. Описывается модель представления маршрута и алгоритм преобразования прямого маршрута в обратный. Приведены результаты экспериментов по имитационному моделированию, описаны натурные эксперименты.

**Ключевые слова:** автономный мобильный робот, навигация муравьев, построение обратного маршрута, задача фуражировки

### Введение

Для мобильных роботов большое значение имеет ориентация в пространстве, запоминание маршрута, способность вернуться в точку отправления и умение повторить маршрут. Среди методов решения этих задач, не требующих построения карты, особое место занимают биологически инспирированные методы ориентации и навигации по аналогии с механизмами ориентации животных, чаще всего – насекомых. В данной работе моделируется поведение муравьев, занимающихся фуражировкой. Этот подход относится к моделированию социального поведения для решения задач групповой робототехники [Карпов, 2016].

Многие исследователи рассматривают принципы навигации муравьев как основу методов навигации и ориентации автономных мобильных роботов. Есть множество вариантов использования отдельных механизмов или морфологических особенностей муравьев: например, роботы, оснащенные специальным компасом для ориентации по поляризации солнечного света ([Lambrinos et al., 2000], [Duponroux et al., 2019]); или имитация феромонного следа [Dorigo and Blum, 2005]. Для следования роботов по маршруту предлагаются методы на основе запоминания моментальных снимков окружающей среды во время изучения маршрута ([Vardy, 2006], [Yu and Kim, 2011]), а также модели векторной навигации на основе искусственной нейронной сети [Goldschmidt et al., 2017], которые используются для запоминания и повторного прохождения

агентами маршрута. Большое внимание уделяется организации памяти для навигации ([Augustine et al., 2012], [Mair et al., 2014]). Каждый из упомянутых методов обладает некоторыми недостатками: одни не учитывают визуальные ориентиры, играющие большую роль в ориентации насекомых; другие требуют построения топологической карты; третьи не реализуют возвращение в исходную точку маршрута.

Ранее автором был разработан метод ориентации без построения карты и без использования средств глобальной навигации [Карпова, 2022]. Метод включает механизм запоминания и интерпретации описания маршрута, позволяющий повторно пройти по запомненному маршруту. Недостатки этого метода – отсутствие возможности вернуться в точку отправления и сложности с обходом препятствий. На текущем этапе исследований разработан алгоритм преобразования прямого маршрута в обратный, а перемещение по маршруту включает обход препятствий.

**Замечание о терминах.** Объект исследования – искусственный автономный агент, действующий в виртуальной или реальной среде и имитирующий поведение живого организма [Wilson, 1987], поэтому в тексте чаще используется термин «анимат». Если речь идет о проведении натуральных экспериментов, то употребляется термин «робот».

## 1. Алгоритмы ориентации

Разработанный метод опирается на принципы фуражировочного поведения рыжих лесных муравьев (*Formica rufa*). Фуражировка включает поиск ресурса, перенос его в гнездо (возвращение в точку отправления) и повтор этого пути. Во время поиска муравей-разведчик запоминает свое положение относительно солнца, визуальные ориентиры, мимо которых он проходит, и примерное расстояние до них ([Длусский, 1967], [Stuse and Wehner, 2011]). С помощью этой информации разведчик может вернуться в гнездо. Затем он мобилизует фуражиров и ведет их к ресурсу, или передает им сведения о запомненном маршруте, и фуражиры самостоятельно идут за ресурсом [Захаров и др., 2013]. Подробнее особенности поведения муравьев описаны в [Карпова, 2021].

Маршрут может быть представлен как последовательность отрезков, на которых анимат движется прямо. В начале каждого отрезка анимат-разведчик запоминает сцену, выбирает основной ориентир и направление движения относительно этого ориентира. Он передвигается в выбранном направлении, и после обхода ориентира переходит к следующему отрезку пути. Если он не видит ни одного ориентира, то поиск считается неудачным, и анимат возвращается на «базу». Если он видит искомый ресурс, то этот ресурс (объект) является для него основным ориентиром сцены. Общий алгоритм действий анимата-разведчика приведен на Рис. 1.

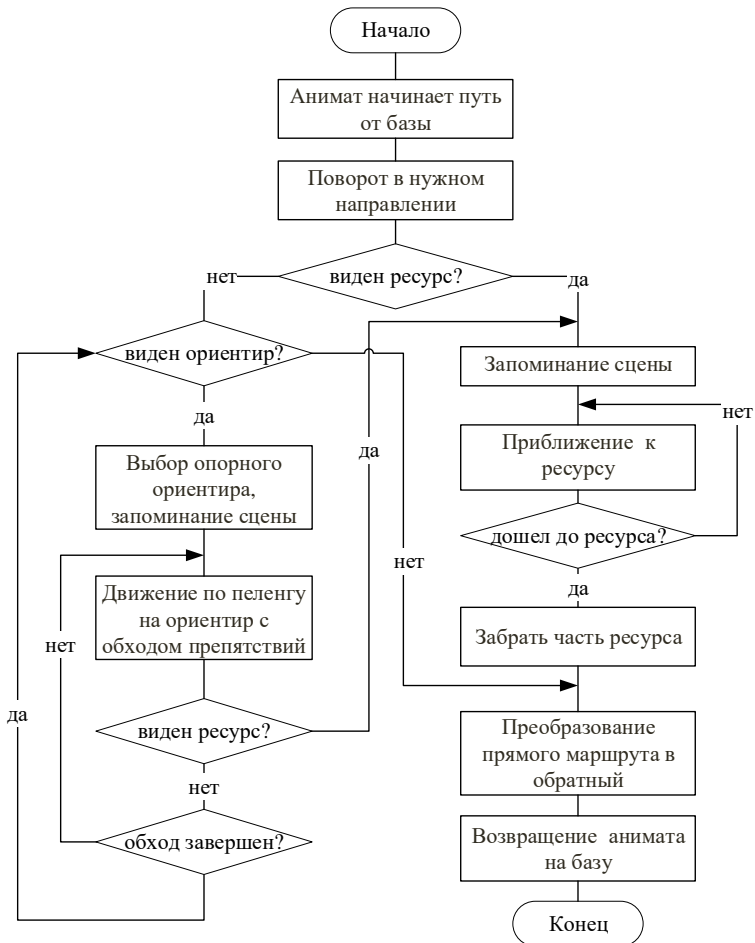


Рис. 1. Общий алгоритм действий анимата-разведчика при поиске ресурса

Далее анимат действует как фуражир и по похожему алгоритму (Рис. 2). Он не ищет новый ориентир, а сопоставляет сцены запомненного маршрута с тем, что видит вокруг себя. При успешном сопоставлении сцен фуражир обходит опорный ориентир, используя процедуру движения по пеленгу с обходом препятствий, и переходит к следующей сцене. При неудаче запускает процедуру поиска опорного ориентира в текущей или следующей сцене, аналогично муравью, потерявшему ориентир. Если фуражир не видит ни одного ориентира, он пытается вернуться на «базу».

На «базу» анимат возвращается практически по тому же алгоритму (Рис. 2) с двумя отличиями: (1) вместо искомого ресурса анимат ищет «базу» и (2) при потере ориентации анимат продолжает поиск «базы»,

продвигаясь в том направлении, где предположительно находится «база». Это обеспечивает унификацию поведенческих процедур.

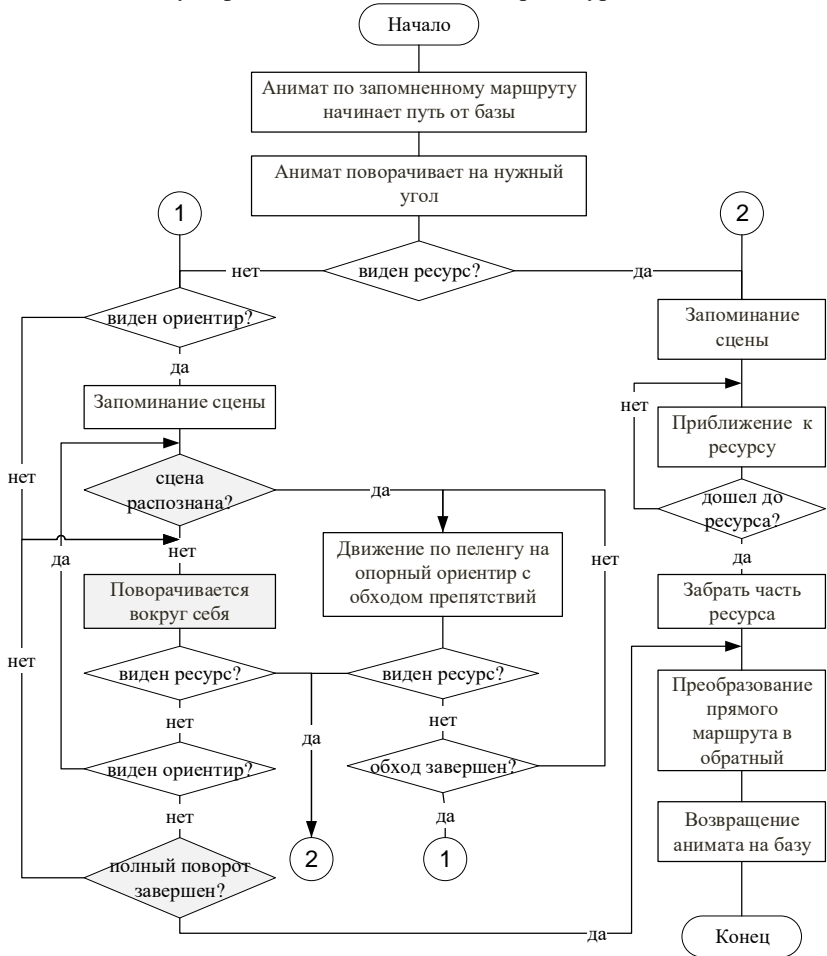


Рис. 2. Общий алгоритм действий анимата-фуражира при повторе пути

## 2. Мир анимата

Определим, как можно представить мир анимата, и какими возможностями он должен обладать для реализации такого поведения. На полигоне, по которому перемещаются аниматы, расположена «база», неподвижные объекты (ориентиры) и некоторые целевые объекты – «ресурс», который нужно перенести на «базу» (Рис. 3). Путь – это последовательность перемещений анимата от одного целевого объекта к

другому (целевые объекты – «база» или «ресурс»). Анимат может двигаться вперед и назад, останавливаться, поворачивать направо и налево. С помощью локатора он «видит» окружающие объекты внутри некоторой области видимости (сектор круга на Рис. 3).

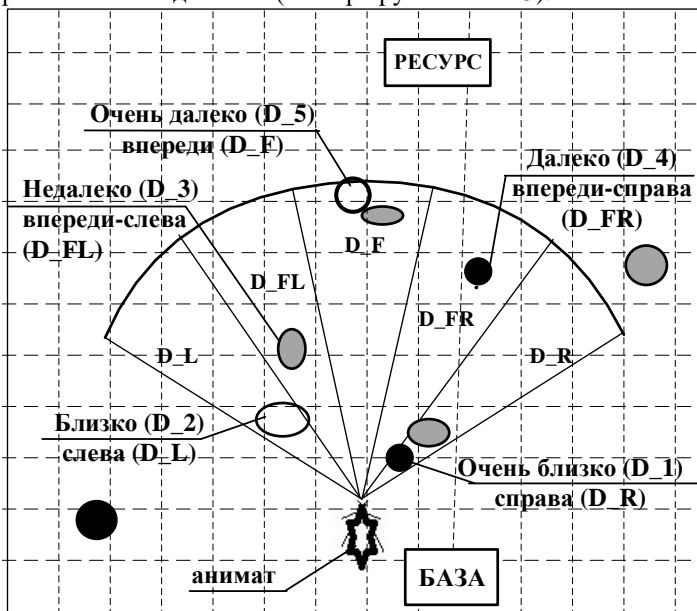


Рис. 3. Представление мира анимата: область видимости, направление, расстояние

Мир анимата должен быть представлен так, чтобы можно было при описании модели оперировать относительными категориями, без привязки к абсолютным физическим величинам. Для этого полигон виртуально разбивается на клетки, размер которых определяется характерным линейным размером робота (модельным или реальным) (Рис. 3), а для понятий «расстояние» и «направление» вводятся порядковые шкалы и правила пересчета реальных значений в условные единицы (клетки).

### 3. Описание маршрута

Пусть в мире анимата существует множество  $\Omega$  из  $N$  распознаваемых им объектов  $x_n$ :  $\Omega = \{x_n\}$ ,  $n=1, \dots, N$ . Под *ориентиром*  $Q$  будем понимать один или несколько объектов, образующих компактную группу, т.е. расположенных близко друг к другу:

$$Q = \{x_i\} (\{x_i\} \subset \Omega): \forall x_k \in Q \exists x_m: x_m \in Q, \rho(x_k, x_m) \leq \Delta \quad (3.1)$$

Здесь  $\rho$  – расстояние между объектами (например, евклидово), а  $\Delta$  – верхняя граница диапазона «очень близко». Наличие одинаковых ориентиров приводит к необходимости распознавания сцен, а не

отдельных ориентиров или объектов [Московский, 2015]. Сцена является совокупностью видимых ориентиров, пространственных отношений между ними, направления по компасу и временной составляющей. При этом сцена определяет поведение анимата и изменение его состояния.

Маршрут описывается как последовательность сцен  $\Omega = \{S_j\}$ ,  $j=1, \dots, L_p$ , где  $L_p$  – длина маршрута. Каждая сцена  $S_j$  состоит из множества ориентиров  $\{O_i^j\}$ , один из которых находится в фокусе внимания и является опорным (относительно него совершаются все действия). Каждый ориентир  $O_i^j$ , в свою очередь, состоит из множества наблюдаемых объектов с их характеристиками (контекстом)  $V$ :  $O_i^j = \{V_k\}$

$$V_k = \{Id, X, \{(dir_i, dist_i)\}, Compas, Timer\}, \quad (3.2)$$

где  $Id$  – идентификатор объекта (например, цвет);  $X$  – признак опорного ориентира (направление обхода ориентира);  $\{(dir_i, dist_i)\}$  – множество пар, хранящих направление и расстояние до левой границы, центра объекта и правой границы соответственно;  $Compas$  – значение датчика «компас»;  $Timer$  – номера тактов в начале и в конце отрезка пути.

Для сопоставления сцен используются правила псевдофизической (пространственной) логики [Pospelov, 1987] и сравнение ориентиров с учетом контекста (подробнее см. [Карпова, 2022]).

#### 4. Преобразование прямого маршрута в обратный

Для возвращения на базу описание маршрута  $\Psi$  преобразовывалось в обратный маршрут  $\Psi'$ . Общий принцип преобразования описания маршрута  $\Psi$  в обратный маршрут  $\Psi'$  заключается в следующем. Обратный маршрут состоит из тех же сцен, что и прямой, но в обратном порядке. Для прямого маршрута конечная точка – искомый ресурс, для обратного – «база». Ориентиры каждой сцены зеркально отображаются слева направо, направление движения меняется на противоположное. При этом короткие сцены пропускаются: это повышает устойчивость анимата (и робота) на маршруте. Алгоритм преобразования прямого маршрута (DirectRoute) в обратный (BackRoute):

```
BackRoute := [] # инициализация обратного маршрута пустым списком
Base := [сцена из одного ориентира – «база»]
Минимальное время нахождения на участке := 50
current_time := (Текущее модельное время)
период_2 := (период времени на последнем участке маршрута)
направление_2 := (направление движения на последнем участке м-та)
Цикл по Line из (DirectRoute) # в обратном порядке
    # Начиная с предпоследнего участка, т.к. последняя сцена не будет
    # видна после разворота анимата в противоположном направлении
    Scene := [] # инициализация новой сцены описания маршрута
    направление_2 := (развернуть на 180°)
```

```

направление_1 := (направление движения на участке Line)
период_1 := (период времени на участке Line)
time = current_time + период_2
Если (период_2) < (Минимальное время нахождения на участке),
то направление_2 := направление_1 # запомнить направление
    период_2 := период_1           # запомнить период
    current_time := time           # запомнить новое текущее время
continue                          # пропустить «короткую» сцену

```

к.е.

**Цикл** по Landmark из (Line) # в обратном порядке – слева направо  
направление\_БО = (направление обхода базового ориентира)

Если направление\_БО = *слева*, то направление\_БО := *справа*  
иначе-если направление\_БО = *справа*, то направление\_БО := *слева*

к.е.

Orientir := [] # инициализация нового ориентира пустым списком

**Цикл** по Object из (Landmark) # в обратном порядке - слева направо  
Добавить Object к Orientir

Пересчитать границы видимости для Object

**к.п.** по Object

# Для Orientir установить новые направления и период:

Установить направление обхода баз.ориентира (направление\_БО)

Установить направление движения на участке (направление\_2)

Установить период движения на участке (период\_2)

Добавить Orientir к Scene

**к.п.** по Landmark

направление\_2 := направление\_1 # запомнить направление

период\_2 := период\_1 # запомнить период

current\_time := time # запомнить новое текущее время

Добавить Scene к Route

**к.п.** по Line

# добавить конечную точку обратного маршрута – «базу»

направление\_2 := (развернуть на 180°)

Установить в Base направление движения на участке (направление\_2)

Установить в Base период движения на участке (период\_2)

Добавить Base к BackRoute

## 5. Организация системы управления анимата

Система управления (СУ) поведением анимата состоит из конечных автоматов с выходом – автоматов Мили. Архитектура СУ является трехуровневой, что позволяет разделить уровни управления по иерархическому принципу. Автоматы нижнего уровня реализуют базовые поведенческие процедуры, управляющие движением анимата, и

некоторые служебные действия. Мета-автоматы отвечают за последовательности действий анимата, например, обойти препятствие. Выходом мета-автоматов является запуск автоматов нижнего уровня. Мета-мета-автомат управляет поведением в целом, его задача – запускать мета-автоматы. Эта архитектура позволяет относительно просто изменять поведение анимата без изменения его структуры и базовых возможностей. Более подробно архитектура СУ описана в [Карпова, 2022].

## 6. Результаты экспериментов

### 6.1. Результаты имитационного моделирования

Имитационное моделирование проводилось на задаче фуражировки. Сначала анимат выходил на поиск ресурса, затем возвращался на «базу» и сам повторял этот путь по запомненному маршруту (4 прохода за одно моделирование). Если в процессе поиска возникала ситуация, при которой анимат не видел ориентиров, он возвращался на «базу» и начинал поиск заново. Примеры запуска моделирующей программы приведены на Рис. 4 (шестиугольник изображает «базу», круг – искомый ресурс).

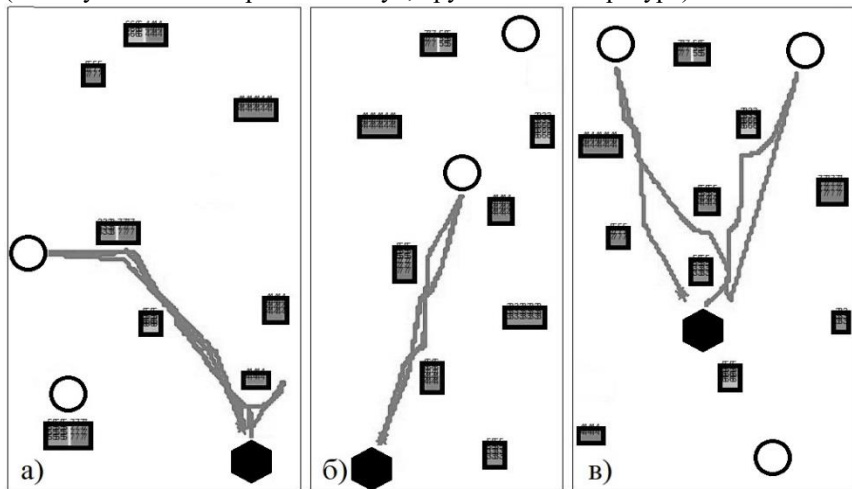


Рис. 4. Примеры маршрутов: а) маршрут с резким поворотом; б) прямой маршрут; в) маршрут, который не удалось повторить

Эксперименты проводились в системе многоагентного моделирования Kvozum, созданной в НИЦ «Курчатовский институт» [Карпов и др., 2018]. Полигон – область  $200 \times 200$  клеток, время моделирования – 20000 тактов для каждого эксперимента. Исследовались различные конфигурации ориентиров на полигоне, и на каждой конфигурации проводилось 5 запусков в разных направлениях (всего – 50 экспериментов). Статистика моделирования приведена в Табл. 1.



Табл. 1. Результаты имитационного моделирования

Кол-во экспериментов	Описание	Кол-во проходов в одном эксперименте	Кол-во возвратов в одном эксперименте	Общее кол-во неудач	Общее кол-во успешных проходов
35	нашел ресурс, вернулся, пошел 2-й раз, вернулся	4	2	0	70
3	нашел ресурс не сразу, вернулся, пошел 2-й раз, вернулся	от 6 до 12	от 3 до 6	0	12
5	нашел ресурс, вернулся, пошел еще раз, не нашел ресурс, не вернулся	3	1	5	5
2	Нашел ресурс, вернулся, 2-й раз не нашел, но вернулся	3	2	2	7
5	не нашел ресурс, вернулся, ходил повторно на поиски	от 14 до 48	от 7 до 24	2	87

Если не учитывать эксперименты, в которых анимат все время возвращался из-за того, что не видит ориентиров, то из 101-го прохода туда-обратно анимат не вернулся 7 раз (примерно 93% успешных проходов). Если считать полностью успешными эксперименты, в которых анимат дважды дошел до одного и того же ресурса, то таких было 38: это 84% от 45 экспериментов, в которых конфигурация ориентиров позволяла анимату успешно решить задачу.

## 6.2. Результаты натуральных экспериментов

Натурные эксперименты проводились на мобильных платформах – мини-роботах серии YARP. Мини-робот – это платформа с дифференциальным приводом, оснащенная дальномерами, гироскопом, камерой и бортовым вычислителем Raspberry Pi 4. Экспериментальный комплекс представляет собой мобильную платформу, удаленный управляющий компьютер и полигон, на котором расположены ArUco-маркеры. Комплекс работает под управлением ROS (Robot Operating System). На бортовом вычислителе развернуты компоненты ROS, отвечающие за выполнение двигательных функций и обработку данных с камеры. Камера – основной источник сведений о наблюдаемых роботом объектах – ориентирах с нанесенными ArUco-маркерами. В экспериментах участвовали роботы на двух платформах серии YARP – платформе YARP-132 и YARP-133. Эти платформы имеют одинаковую архитектуру

системы управления, но отличаются по габаритам и скоростям движения. Несмотря на это, итоговые результаты их работы по запоминанию и обработке маршрутов были идентичными. Это свидетельствует о независимости разработанной поведенческой модели от физической реализации. Для экспериментов использовался полигон Лаборатории робототехники «Курчатовского института» (Рис. 5).



*Рис. 5. Натурные эксперименты на полигоне (YARP-133)*

При переносе моделирующей программы на реального робота была произведена калибровка камеры и настройка скоростей движения робота, а также написан специальный модуль для перевода данных камеры в формат системы Kvozum. Для исключения ложных срабатываний камеры на полигоне было введено ограничение на номера распознаваемых маркеров (от 1 до 9). Никаких других изменений в управляющую программу вносить не потребовалось. Таким образом, одна и та же программа использовалась и для имитационных экспериментов, и для управления роботом на реальном полигоне. В ходе натурных экспериментов роботы вели себя абсолютно аналогично аниматам.

## **Заключение**

Сравнение полученных результатов с аналогичными работами вызывает определенные сложности. Условия проведения экспериментов сильно отличаются, поэтому сравнение статистических показателей было бы некорректным. Большинство работ по данной тематике апеллирует к точности прохождения маршрута [Lambrinos et al., 2000]; в данном же исследовании точность не играет существенной роли. Вместо панорамных камер (как в [Augustine et al., 2012], [Yu and Kim, 2011] и др.) наши роботы имели угол развертки камеры 120°. Поэтому сравнивать с аналогами можно только по качественным показателям.

По сравнению с работами, которые ограничивались имитационным моделированием, данный метод имеет преимущества, т.к. он проверен на

различных робототехнических платформах и в реальных условиях. По отношению к тем работам ([Lambrinos et al., 2000], [Dall'Osto et al., 2021]), в которых роботы действуют в реальной среде, данный метод несколько проигрывает, т.к. для него необходимы определенные ориентиры, которые он умеет распознавать. Переход к произвольным ориентирам повлечет усложнение процедуры распознавания и увеличение объема памяти для хранения описания маршрута. Если же говорить о методе в целом, то к его достоинствам можно отнести невысокие требования к точности позиционирования, отсутствие требования уникальности ориентиров, работу с камерой с небольшим углом развертки, обход препятствий и возможность возвращения в точку отправления.

**Благодарности.** Автор выражает благодарность сотрудникам лаборатории робототехники НИЦ «Курчатовский институт», на базе которого проводились эксперименты, и, в первую очередь, руководителю лаборатории Карпову В.Э.

### Список литературы

- [Augustine et al., 2012] Augustine M., Ortmeier F., Mair E., Burschka D., Stelzer A., Suppa M. Landmark-Tree map: A biologically inspired topological map for long-distance robot navigation. // IEEE International Conference on Robotics and Biomimetics, ROBOT 2012 – Conference Digest.
- [Cruse and Wehner, 2011] Cruse H., Wehner R. No need for a cognitive map: Decentralized memory for insect navigation // PLoS Comput. Biol., 2011, № 7(3).
- [Dall'Osto et al., 2021] Dall'Osto D., Fischer T., Milford M. Fast and Robust Bio-Inspired Teach and Repeat Navigation. // 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
- [Dorigo and Blum, 2005] Dorigo M., Blum C. Ant colony optimization theory: A survey. // Theoretical Computer Science, 2005, № 344(2-3).
- [Dupeyroux et al., 2019] Dupeyroux J., Viollet S., Serres J.R. An ant-inspired celestial compass applied to autonomous outdoor robot navigation. // Robotics and Autonomous Systems, 2019, № 117.
- [Goldschmidt et al., 2017] Goldschmidt D., Manoonpong P., Dasgupta S. A neurocomputational model of goal-directed navigation in insect-inspired artificial agents. // Frontiers in Neurorobotics, 2017, № 11(APR).
- [Lambrinos et al., 2000] Lambrinos D., Möller R., Labhart T., Pfeifer R., Wehner R. A mobile robot employing insect strategies for navigation. // Robotics and Autonomous Systems, 2000, № 30(1).
- [Mair et al., 2014] Mair E., Augustine M., Jäger B., Stelzer A., Brand C., Burschka D., Suppa M. A biologically inspired navigation concept based on the Landmark-Tree map for efficient long-distance robot navigation. // Advanced Robotics, 2014, 28(5).
- [Pospelov, 1987] Pospelov D.A. Fuzzy reasoning in pseudo-physical logics. // Fuzzy Sets and Systems, 1987, № 22 (1–2).
- [Vardy, 2006] Vardy A. Long-range visual homing. // IEEE International Conference on Robotics and Biomimetics, ROBOT 2006.
- [Wilson, 1987] Wilson S.W. Classifier Systems and the Animat Problem. // Machine Learning, 1987, № 2(3).

- [Yu and Kim, 2011] Yu S.E., Kim D. Image-based homing navigation with landmark arrangement matching. // Information Sciences, 2011, № 181(16).
- [Длусский, 1967] Длусский Г.М. Муравьи рода Формика. – Наука, 1967.
- [Захаров и др., 2013] Захаров А.А., Длусский Г.М., Горюнов Д.Н., Гилев А.В., Зрянин В.А., Федосеева Е.Б., Гороховская Е.А., Радченко А.Г. Мониторинг муравьев Формика. – КМК, 2013.
- [Карпов, 2016] Карпов В.Э. Моделирование социального поведения в групповой робототехнике. // Управление большими системами, № 59.
- [Карпов, 2011] Карпов В.Э. Об одном механизме реконструкции схем. // VI Международная научно-практическая конференция “Интегрированные модели и мягкие вычисления в искусственном интеллекте”, 2011. Сб. науч. трудов. В 2-х томах. Т.1, М.: Физматлит.
- [Карпов и др., 2018] Карпов В.Э., Ровбо М.А., Овсянникова Е.Е. Система моделирования поведения групп робототехнических агентов с элементами социальной организации Кворум. // Программные продукты и системы, 2018, № 31(3).
- [Карпова, 2021] Карпова И.П. Организация маршрута анимата на основе визуальных ориентиров и распознавания сцен. // Мехатроника. Автоматизация. Управление, 2021, 22(10).
- [Карпова, 2022] Карпова И.П. Об одном биоинспирированном подходе к ориентации роботов, или настоящий «муравьиный» алгоритм. // Управление большими системами, 2022, № 96.
- [Московский, 2015] Московский А.Д. Метод распознавания сцен для задачи навигации мобильных роботов. // Мат. II Всерос. научно-практ. семинара «Беспилотные транспортные средства с элементами искусственного интеллекта (БТС-ИИ-2015)» (9.10.2015г., г. СПб, Россия): Политехника-сервис.

## ABOUT ROBOTS ORIENTATION BASED ON VISUAL LANDMARKS AND COMPASS

I.P. Karpova (*karpova\_ip@mail.ru*)  
National Research University Higher School of Economics,  
Moscow

The paper considers a method of using visual landmarks to memorize the path traveled by a mobile robot (animat), based on the mechanism of ant navigation. A route representation model and an algorithm for converting a direct route into a reverse route are described. The results of simulation experiments and experiments with real robots confirm the efficiency of the method.

**Keywords:** autonomous mobile robot, ant navigation, reverse route, foraging task