



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 15 за 2015 г.



Давыдов О.И., [Платонов А.К.](#)

Сеть Пассфреймов -  
комбинированная модель  
операционной среды  
мобильного робота

**Рекомендуемая форма библиографической ссылки:** Давыдов О.И., Платонов А.К. Сеть Пассфреймов - комбинированная модель операционной среды мобильного робота // Препринты ИПМ им. М.В.Келдыша. 2015. № 15. 28 с. URL: <http://library.keldysh.ru/preprint.asp?id=2015-15>

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

**О.И. Давыдов, А.К. Платонов**

**Сеть Пассфреймов —  
комбинированная модель  
операционной среды  
мобильного робота**

**Москва — 2015**

**Давыдов О.И., Платонов А.К.**

**Сеть Пассфреймов - комбинированная модель операционной среды  
мобильного робота**

В работе рассматривается структура внутреннего представления информации для управления движением автономного мобильного робота РБ-2. Предлагается модель операционной среды мобильного робота, которая представляет собой комбинацию метрической, топологической и семантической моделей. Вводится понятие Пассфрейма как ключевого элемента модели, описывается его структура и основные свойства. Рассматривается структура модели операционной среды в виде сети Пассфреймов. На данных, получаемых с лазерного сканера, описывается алгоритм построения отдельных Пассфреймов, а также алгоритм формирования сети в целом. Описывается процедура планирования маршрута движения робота из исходного положения к целевому положению. Все алгоритмы, предлагаемые в данной работе, реализованы в виде программного обеспечения системы управления движением роботом РБ-2.

**Ключевые слова:** мобильный робот, пассфрейм, лазерный сканер, модель операционной среды, алгоритмы планирования маршрута, робот РБ-2.

**Oleg Izmailovich Davydov, Alexander Konstantinovich Platonov**

**Passframe Network - combined operating environment model  
for a mobile robot**

The paper describes an internal information presentation for controlling movements of an autonomous mobile robot RB-2. It proposes a model of mobile robot operating environment that combines metric, topological and semantic models. The model proposes a term Passframe as a key element and describes its structure and main features. An operating environment model's structure is presented as a network of Passframes. Data received from a laser scanner provides input to an algorithm for building individual Passframes as well as the network as a whole. This paper describes a procedure for planning robot's movements between a starting point and a target. All motion planning algorithms, presented in this paper, have been implemented in RB-2's movement control software.

**Keywords:** mobile robot, Passframe, laser scanner, operational system model, motion planning algorithm, robot RB-2.

**Работа поддержана грантом РФФИ № 15-08-06431**

## 1. Введение

Предлагаемая работа проводится в рамках исследований, цель которых – интеллектуализация робототехники, повышение автономности и адаптивности робототехнических систем, разработка недорогих, но эффективных алгоритмов безопасного управления автономными и полуавтономными мобильными роботами. По своему назначению рассматриваемые мобильные роботы должны иметь подвижное основание и должны уверенно перемещаться в закрытых помещениях в окружении не очень большого числа нормально движущихся людей: в бытовых условиях городских квартир и в общественных местах, таких, как больницы, школы, офисы, музеи.

Управление движением – ключевая задача, которая в мобильном роботе должна эффективно решаться. Фактически, успех решения всех остальных задач функционирования робота во многом зависит от решения этой задачи. Система управления мобильного робота должна содержать информацию об операционной среде, в которой он перемещается и функционирует. Даже в условиях слабо меняющегося состояния операционной среды для выработки конкретных решений система управления должна обладать достаточно большим объемом разнородных данных. Выбор способа внутреннего представления этой информации, соответствующего задачам движения, лежит в основе построения системы управления мобильным роботом.

### **1.1. Основные подходы к созданию модели операционной среды**

Под «операционной средой робота» ниже понимается ограниченное по размеру подмножество учитываемых обстоятельств из его окружения, достаточное для решения задачи его функционирования. Операционная среда у мобильного робота – это структура внутреннего представления сенсорной информации, доступной системе управления, используемая для автономного или полуавтономного решения его двигательных задач.

Проблеме построения моделей операционной среды мобильного робота посвящено множество работ. Для тех, кого интересует история этой ключевой проблемы робототехники, отметим, что первые реальные попытки Лисп-ориентированной логической организации движения по данным видеокамеры (робот «Шейки» [1]) довольно быстро сменилась работами по использованию процедурных преобразований более удобной информации, получаемой от сканирующего лазерного дальномера ([2], [3]). В частности, в работах [4], [5] и [6] в такой постановке было введено в рассмотрение и подробно исследовалось специальное абстрактное пространство (оно получило имя "терраин") – в виде множества координат области передвижения с отношением их взаимной видимости в процессе дальнометрии. Этот подход активно развивается ниже в целях построения алгоритмической модели операционной среды мобильного робота с дальномером.

В современной научно-технической литературе представлены три основных типа модели операционной среды мобильного робота:

- метрическая, координатная;
- топологическая, сетевая;
- семантическая, предикатная.

Все три типа модели имеют свои преимущества и недостатки – с позиций эффективности их конкретного применения. Они широко исследованы в литературе [7], [8], [9], [10], [11], [12] и [13].

*Метрические модели* обычно в том или ином виде используют координаты поверхностей препятствий в операционной среде, окружающей робота. Такие модели при известных координатах текущего и целевого положений робота используются для формирования промежуточных координат траектории движения робота. Как правило, метрическое представление дает возможность с определённой точностью позиционировать робот и строить оптимальную траекторию движения. Однако большой объем используемой при этом информации делает такие модели не очень удобными даже для описания слабо динамических сред. Для метрических моделей разработано большое число алгоритмов, которые применяются, в основном, в статических средах.

Координатное описание траектории, как правило, представляет собой последовательность координат ряда промежуточных положений робота (подцелей) при его движении от исходной позиции к целевой позиции. В системах управления мобильными роботами координатное описание траектории движения обычно является основой для выработки управляющих сигналов, которые подаются на исполнительные элементы – движители робота.

*Топологические модели*, как правило, описывают операционную среду в виде сети. Узлами сети обозначаются некоторые известные позиции в операционной среде, а дуги такой сети определяют свойство взаимной достижимости узлов, т.е. возможности роботу переместиться из одной позиции в другую. Позиции – узлы сети – обычно представлены именами-идентификаторами. Топологические модели дают возможности создавать цепочки поименованных позиций робота, которые называются маршрутом движения робота от заданного исходного узла сети к целевому узлу. Если узлам сети поставлены в соответствие координаты соответствующих позиций, то на основе выбранного маршрута можно строить траектории, состоящие из цепочек координат промежуточных подцелей. Такие комбинированные модели устраняют необходимость в метрической модели операционного пространства робота. Они все чаще используются в системах оперативного планирования и управления роботом.

*Семантические модели* описывают операционную среду в виде набора предикатных выражений, т.е. логических отношений между отдельными параметрами среды. Семантические модели используются для построения умозаключений, которые позволяют решать наиболее общие, смысловые задачи

построения движения. Например, расшифровать задание, сформулированное человеком-оператором, определить набор целевых предикатов и, используя их логические преобразования, спланировать требуемое поведение робота в виде последовательности целей для работы различных управляющих подсистем робота вместе с последовательностью целей обработки данных, поступающих с различных групп его датчиков.

В принципе, семантические модели тоже позволяют выбирать маршрут движения, но топологические модели для решения таких задач более эффективны. Дело в том, что семантические модели абсолютно не приспособлены для формирования координатной траектории маршрута движения робота, это может потребовать огромных усилий для написания соответствующих алгоритмов. Семантические модели более приспособлены и предназначены для обработки разнородной информации, поступающей с разных источников, для приведения этой информации в предикатную форму и формирования фактов и предположений об операционной среде. В результате обработки этих фактов и предположений с помощью правил логического вывода формируется ряд предикатных условий достижения выбираемых целей поведения, передаваемых на исполнение управляющим системам робота.

В данной работе представлена попытка разработки комбинированной модели операционной среды, которая включала бы в себя все три типа модельного представления и которая могла бы использоваться одновременно и для построения логических выводов и умозаключений, и для поиска цепочек подцелей на пути их достижимости, и для формирования координатной траектории движения.

## **1.2. Мобильный робот РБ-2**

Комбинированная модель операционной среды разрабатывается для управления движением мобильного робота РБ-2. Общий вид робота показан ниже на рис. 1. Робот представляет собой подвижную платформу с установленным на ней манипулятором. Подвижная платформа робота оборудована дифференциальным приводом с двумя передними ведущими колесами и двумя задними поддерживающими омни-колесами, способными с низким трением как вращаться, так и перемещаться в боковых направлениях.

Система управления движением робота РБ-2 получает информацию из нескольких источников. На корпусе робота установлен лазерный сканер, на осях ведущих колес – одометры. Робот оборудован контактными датчиками препятствий, инфракрасными и ультразвуковыми датчиками расстояний. Электроника системы управления робота построена по двухуровневой схеме. На нижнем уровне находятся контроллеры двигателей с усилителями мощности и контроллеры сенсорных устройств. На верхнем уровне стоит компьютер с двухъядерным процессором Intel, с тактовой частотой 1.8ГГц, работающий под управлением операционной системы Windows 7 Embedded.

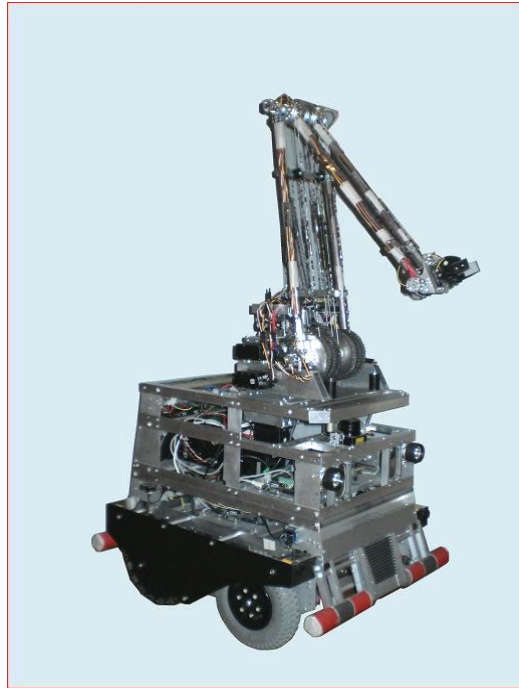


Рис.1. Робот РБ-2

Имеются оперативная память 4Гб, 16Гб внешней флеш-памяти и сенсорный монитор с диагональю 7 дюймов. Связь верхнего уровня электроники с нижним уровнем осуществляется по каналам USB. Робот оборудован системой ручного внешнего радиоуправления, а также датчиком ИК сигналов для управления с помощью ИК пульта.

Программы системы управления роботом РБ-2 имеют гибридную архитектуру, состоящую из трех уровней:

- логическо-декларативного уровня, определяющего поведение робота, содержащего систему оценки ситуации в целом и систему стратегического планирования;
- перцепторно-моторного уровня, выполняющего команды логическо-декларативного уровня и содержащего систему оперативного планирования, управления и обработки сенсорных данных;
- сенсорно-активаторного уровня, выполняющего команды перцепторно-моторного уровня, управляющего приводами, сенсорами и контролирующего робота в нештатных ситуациях.

Автономное программное управление движением подвижной платформы робота осуществляется на основе описываемой ниже комбинированной модели его среды передвижения. Управление движением манипулятора в данной работе не рассматривается.

Программное обеспечение робота РБ-2 написано на языке VisualBasic.Net. Весь массив постоянной информации, необходимой для управления роботом, хранится в базе данных. На данном этапе работ используется платформа Access 2007.

### **1.3. Свойства модели операционной среды робота РБ-2**

Робот РБ-2 предназначен для работы в помещениях – в среде обитания человека. Эта среда является слабо динамической операционной средой. С точки зрения требований к разработке системы управления робота окружающий его мир, который включает в себя и нормально передвигающиеся людей, последовательно и непрерывно меняется, но не с очень большой скоростью и не кардинально. Темп этих изменений позволяет алгоритмам системы управления вовремя реагировать на них, успевать оценивать ситуацию и перестраивать режим движения.

Операционная среда робота состоит преимущественно из статических компонентов. Положение стен помещения, окон, проемов дверей, практически не меняются со временем, а крупная мебель очень редко передвигается с места на место. Для системы управления робота РБ-2 модель статической составляющей операционной среды формируется в ходе специальной процедуры “знакомства” со средой и запоминается в базе данных. В этом смысле мы будем говорить, что роботу становится известна Статическая модель, или Карта операционной среды.

Каждому объекту среды в Статической модели в базе данных ставится в соответствие некоторая величина, характеризующая вероятность изменения положения этого объекта в пространстве. Это сделано потому, что в процессе построения Карты среды в ее состав включаются и полустатические объекты, каковыми, например, являются стулья и кресла. Положение полустатических объектов фиксируется в момент создания Карты, но не исключается возможность изменения этого положения. Алгоритмы системы управления используют эту информацию при планировании движения робота.

Для безопасного управления движением робота Статической модели операционной среды не достаточно. Робот должен располагать оперативной информацией о текущем состоянии среды, о новых препятствиях, возникающих в среде, об изменении положения этих препятствий и о своей позиции в среде. В составе этой информации обязательно должны присутствовать все объекты среды, и статические, и полустатические и динамические. Данные эти должны постоянно обновляться и передаваться на обработку системе управления роботом. В этом смысле мы будем говорить, что роботу известна Оперативная модель среды.

Таким образом, полная модель операционной среды робота РБ-2 включает в себя две модели, Статическую и Оперативную. Обе модели имеют одинаковую структуру и формируются путем обработки данных, поступающих от лазерного сканера.

Статическая модель (Карта) операционной среды робота создается в ходе специальной процедуры её формирования и хранится в базе данных системы управления робота. Эта Карта используется двумя уровнями подсистемы



управления роботом: логическо-декларативным уровнем – для планирования поведения робота, и перцепторно-моторным уровнем – в процессе формирования маршрута движения.

Оперативная модель постоянно, с периодичностью 100 мсек пересоздается в процессе движения робота РБ-2. Перцепторно-моторный уровень использует Оперативную модель для построения требуемой координатной траектории по ходу движения робота. Логическо-декларативный уровень использует эту модель для оценки положения робота, а также и для корректировки поведения робота.

Несмотря на то что обе модели имеют одинаковую структуру и формируются одними и теми же программами, алгоритмы обработки исходных данных в процессе построения Карты среды могут иметь некоторые особенности, связанные с тем, что требуемое поведение робота может отличаться от простого перемещения в операционной среде. В данной публикации вопросы, связанные со случаями построения Карты - Статической модели при специфических особенностях требуемого поведения робота, не рассматриваются.

## 2. Формирование модели операционной среды

Как Статическая, так и Оперативная модели среды робота РБ-2 формируются из однотипных объектов, названных Пассфреймами.

Геометрически Пассфрейм – это замкнутый выпуклый многоугольник на горизонтальной плоскости движения робота, внутреннее пространство которого свободно от препятствий. Многоугольник ограничен массивом векторов. Последовательность векторов в массиве такова, что они обходят область внутри Пассфрейма против часовой стрелки. Векторы задаются начальной и конечной точками. Все векторы в пределах Пассфрейма перенумерованы. Номер вектора является его идентификатором в рамках данного Пассфрейма. Каждому вектору присваивается семантический маркер, определяющий тип данного вектора (см. рис.2).



Рис.2. Соседние Пассфреймы и ссылки между ними

Каждому Пассфрейму присваивается уникальный в рамках данной модели номер-идентификатор, а также семантический маркер, определяющий тип Пассфрейма. Соседние Пассфреймы стыкуются друг с другом, если между ними имеются одинаковые векторы, направленные в противоположные стороны, причем вектор одного Пассфрейма имеет ссылку на соответствующий вектор соседнего с ним Пассфрейма, и наоборот. Благодаря этим ссылкам, массив Пассфреймов образует сеть, узлами которой являются сами Пассфреймы, а дугами – взаимные ссылки векторов друг на друга

Основным источником информации для построения модели среды в виде сети Пассфреймов являются данные лазерного сканера Hokuyo URG-04LX-UG01 (см рис. 3), установленного в передней части робота. Данные предварительно обрабатываются и фильтруются. Предварительная обработка проводится драйвером сканера, который создает массив действительных чисел, расстояний (в сантиметрах) от сканера до поверхности препятствий, находящихся в области видимости сканера. Массив получаемых расстояний упорядочен в порядке возрастания угла сканирования. Угол сканирования меняется от  $-120^\circ$  до  $+120^\circ$  с шагом  $0,35^\circ$ . Всего в массиве 683 измерения.



Рис.3. Сканер Hokuyo UdG-04LX-UG01

Максимальная дальность измерений (граница видимости сканера) составляет 560 см. По паспортным данным, ошибка измерений по углу составляет  $0.001^\circ$ , а по дальности  $\pm 0.1$  см на дистанциях до 100 см., и  $\pm 1\%$  на дистанциях более 100 см. По опыту работы с дальномером угловые погрешности соответствуют паспорту, а в измерениях дальности присутствуют выбросы неверных значений. Кроме того, на дистанциях более 400 см погрешности в измерениях дальности возрастают, примерно до 2-3%

Из-за шума измерений дальностей и наблюдаемых их выбросов в разработанной системе управления РБ-2 получаемые данные фильтруются Медианным фильтром. Для этого их значения внутри окна фильтра сортируются в порядке возрастания (убывания), и значение, находящееся в середине этого

упорядоченного списка, поступает на выход фильтра. Окно перемещается вдоль фильтруемого массива, и такие операции повторяются. Как показал опыт, медианная фильтрация – эффективная процедура обработки данных, уничтожающая подавляющее большинство импульсных помех. В связи с этим она широко применяется в системах обработки изображений [14].

Пример построения модели операционной среды подобного типа представлен ниже на рис. 4. Здесь изображены воображаемый план помещения, робот, находящийся в этом помещении, и модель Пассфреймов, которая должна быть автоматически сформирована системой управления робота в результате обработки данных лазерного сканера.

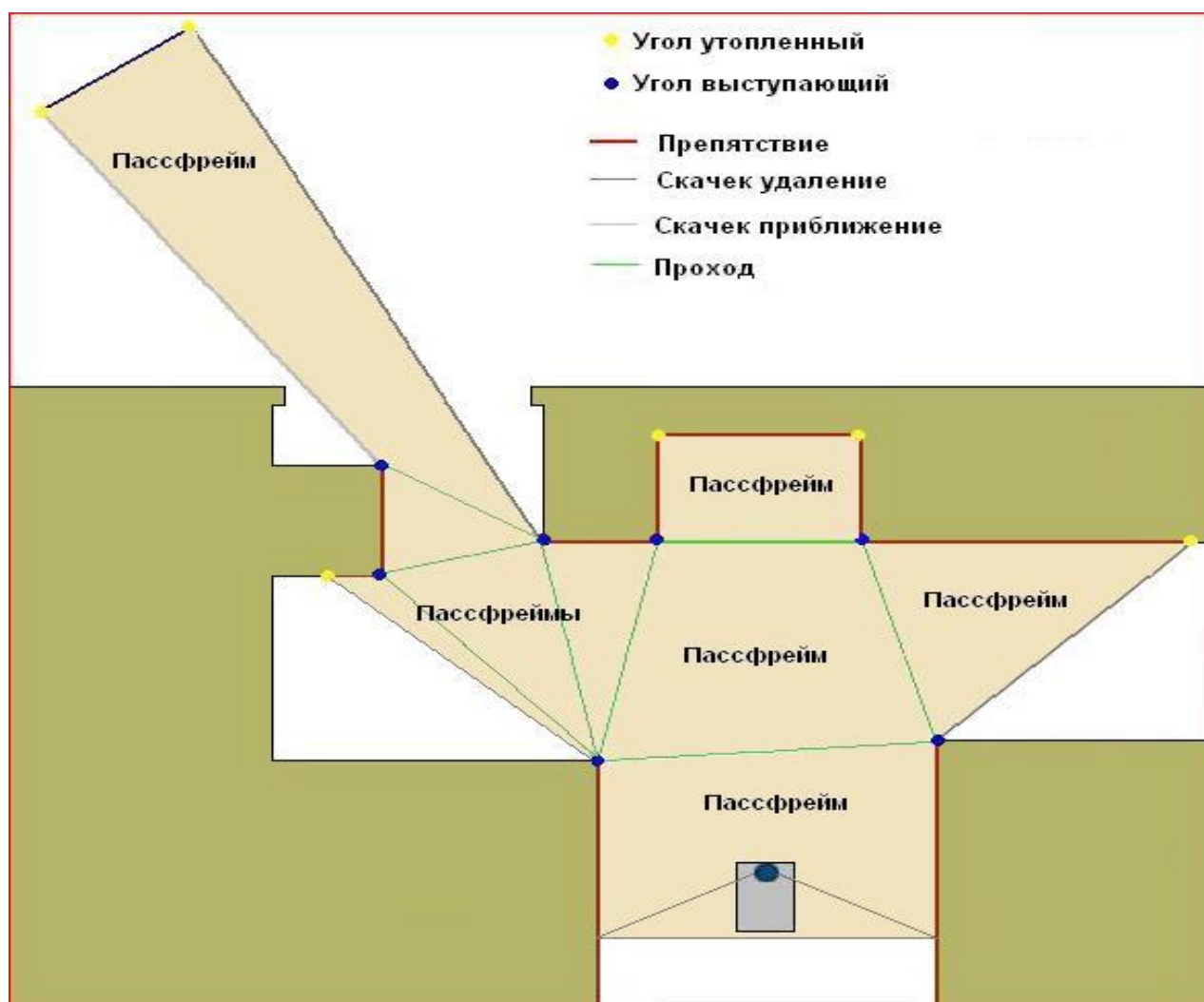


Рис.4. Комбинированная модель операционной среды

Полный процесс построения комбинированной модели среды состоит из трех этапов. На каждом этапе строятся объекты, которым ставятся в соответствие семантические маркеры.

## 2.1. Построение массива семантически помеченных точек

В результате предварительной обработки и фильтрации исходных данных формируется массив точек в полярной системе координат  $(\alpha_i, d_i)$  (где  $\alpha_i$  – угол,  $d_i$  – дистанция), определяющий положение препятствий вокруг робота.

Каждой точке ставится в соответствие один из следующих маркеров:

“Препятствие”;

“Граница видимости” – дистанция за границей «видимости» сканера;

“Скачек удаление” – скачек дальности, удаление;

“Скачек приближение” – скачек дальности, приближение;

“Угол выступающий” – угол, выступающий в сторону робота;

“Угол утопленный” – угол, утопленный, с точки зрения робота.

Идентификация точек типа “Препятствие” и “Граница видимости” производится по значению дистанции для данного измерения. Если дистанция меньше границы видимости сканера, то точке присваивается маркер “Препятствие”, если больше – “Граница видимости”.

### Алгоритм обнаружения скачков дальности

Для идентификации скачков дальности строится массив разности дистанций

$$\Delta D_i = D_{i-1} - D_{i+1}.$$

Из этого массива  $\Delta D_i$  (на рис. 5 он показан красным цветом) выделяются значения, превышающие некоторую пороговую величину

$$\text{abs}(\Delta D_i) > \Delta D_{\text{max}}.$$

Значение константы  $\Delta D_{\text{max}}$  подбирается эмпирически. Для реалий городской квартиры принято  $\Delta D_{\text{max}} = 20\text{см}$ . Соответствующим точкам присваиваются маркеры “Скачек удаление”, если  $\Delta D_i > 0$ , и “Скачек приближение”, если  $\Delta D_i < 0$ .

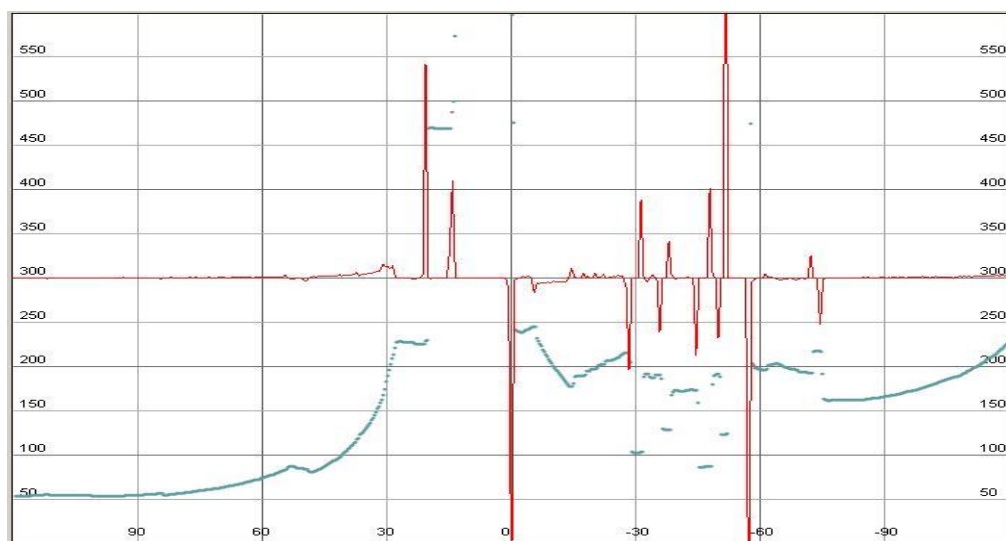


Рис.5. Дистанции и разность дистанций

### Алгоритм обнаружения углов-изломов

Для идентификации изломов препятствий строится массив Абсолютных углов  $\varphi_i$  поверхности препятствий (определение Абсолютного угла показано на рис. 6).

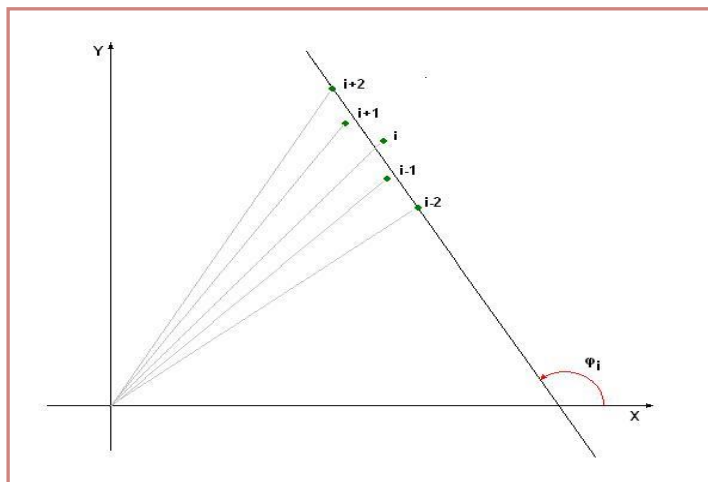


Рис. 6. Абсолютный угол  $\varphi_i$

Далее строится массив разностей Абсолютных углов (см. рис. 7):

$$\Delta\varphi_i = \varphi_{i-1} - \varphi_{i+1}.$$

Из этого массива выделяются значения, превышающие некоторую пороговую величину

$$\text{abs}(\Delta\varphi_i) > \Delta\varphi_{\text{max}}.$$

Значение константы  $\Delta\varphi_{\text{max}}$  подбирается эмпирически. Для реалий городской квартиры принято  $\Delta\varphi_{\text{max}}=40^\circ$ . Соответствующим точкам присваиваются маркеры “Угол выступающий”, если  $\Delta\varphi_i > 0$ , и “Угол утопленный”, если  $\Delta\varphi_i < 0$ .

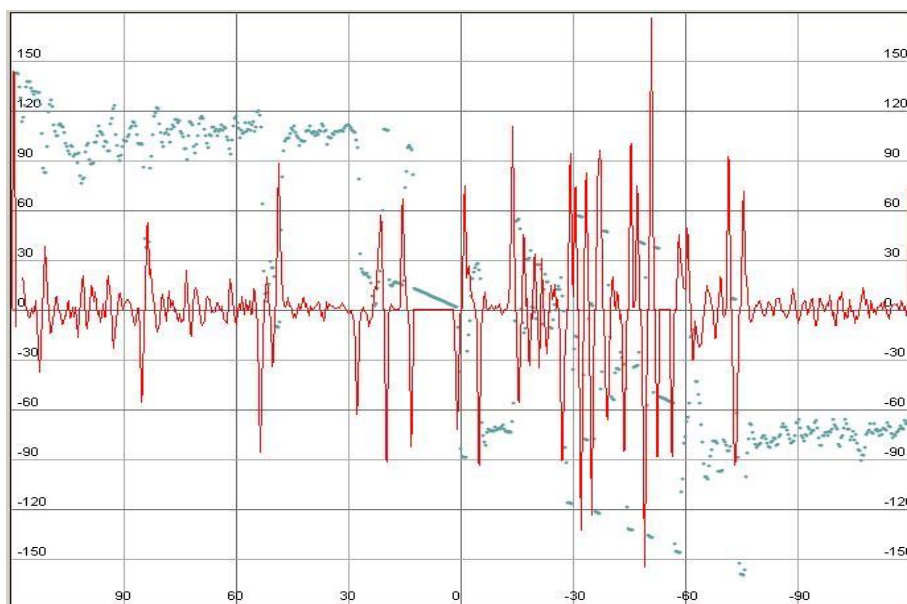


Рис. 7. Абсолютные углы (серые точки) и разность Абсолютных углов (красные линии)

## 2.2. Построение массива семантически помеченных векторов

Массив семантически помеченных точек является исходной информацией для построения массива семантически помеченных векторов. Всем векторам присваивается один из следующих семантических маркеров:

- “Препятствие”;
- “Граница видимости”;
- “Скачек удаление”;
- “Скачек приближение”;
- “Прохода нет”.

В массиве семантически помеченных точек выделяются последовательности стоящих подряд точек, помеченных одинаковыми маркерами типа “Препятствие”, “Граница видимости”, “Скачек удаление” и “Скачек приближение”. Координаты крайних точек каждой последовательности пересчитываются в декартовую систему координат и запоминаются в качестве начальных и конечных точек отрезков. Точки с маркерами “Угол выступающий” и “Угол утопленный” в этой процедуре во внимание не принимаются.

Формируется массив ориентированных отрезков-векторов (см. рис. 8). Отрезки ориентированы в направлении от точки с меньшим углом  $\alpha_m$  к точке с большим углом  $\alpha_k$ , где  $k > m$ . Каждому вектору присваивается номер-идентификатор. В структуре данных каждого вектора запоминаются ссылки на идентификаторы предыдущего и последующего векторов. Термины предыдущий и последующий определяются последовательностью расположения векторов против часовой стрелки относительно позиции сканера.

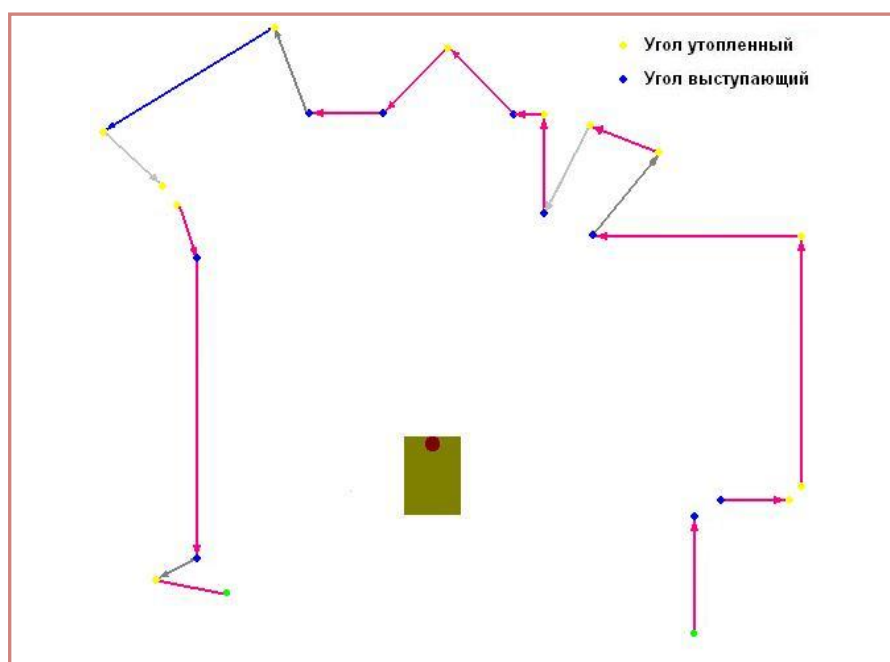


Рис. 8. Пример массива векторов в окружении робота



Каждому вектору присваивается маркер, соответствующий маркеру точек, лежащих под ним. Концы вектора помечаются маркерами точек, которые непосредственно примыкают к концам отрезка. Например, если перед рассматриваемым вектором с маркером “Препятствие” находится точка с маркером “Угол выступающий”, а после вектора – точка с маркером “Скачек удаление”, то начало вектора помечается как “Угол выступающий”, а конец – как “Скачек удаление”.

С геометрической точки зрения, в массиве векторов бывают разрывы, которые образовались из-за игнорирования точек с маркерами “Угол выступающий” и “Угол утопленный”. Эти разрывы уничтожаются следующим образом.

Если расстояние между концом предыдущего вектора и началом следующего вектора больше некоторой минимально допустимой величины  $\Delta d_{\min}$ , подбираемой эмпирически, то между этими векторами вставляется новый вектор, как это показано на рис. 9. Вставляемому вектору присваивается маркер “Препятствие”, а его начальной и конечной точке присваиваются маркеры “Угол выступающий” или “Угол утопленный” в соответствии с маркерами точек, лежащих под ним.

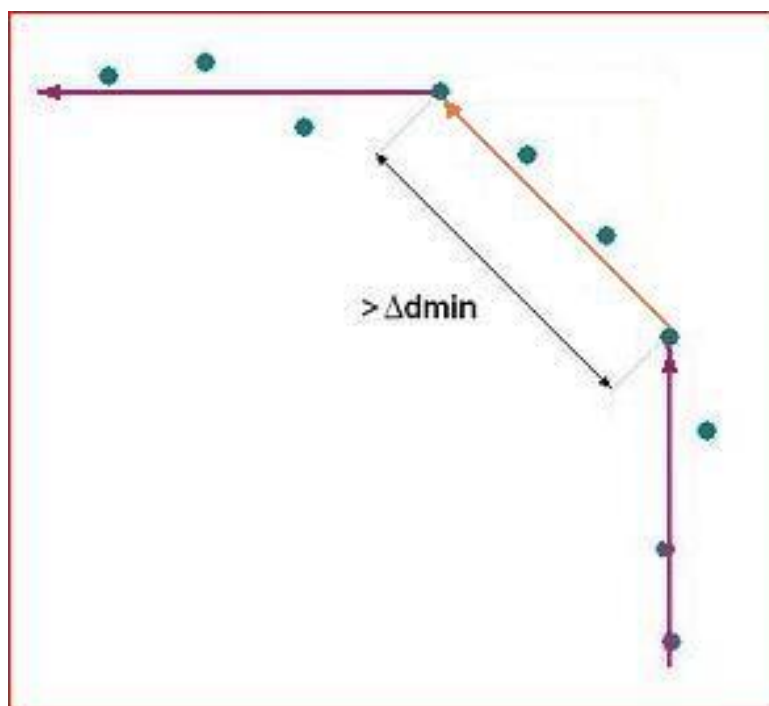


Рис. 9. Вставка вектора

Если расстояние между концом предыдущего вектора и началом следующего вектора меньше  $\Delta d_{\min}$ , то производится попытка удлинения обоих отрезков до точки их пересечения (рис. 10).

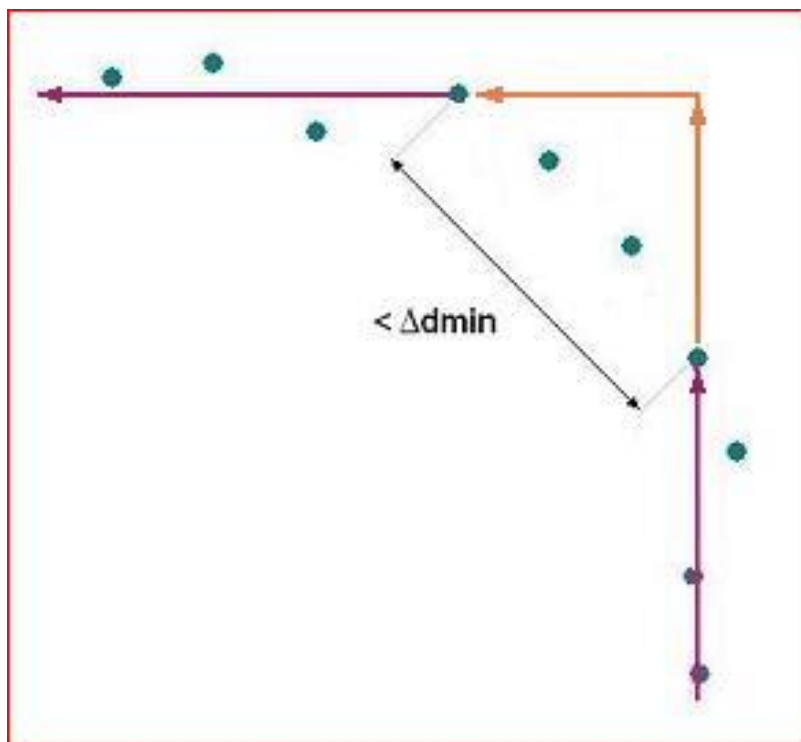


Рис. 10. Удлинение векторов

Если же расстояние между концом предыдущего вектора и началом следующего вектора больше допустимой величины  $\Delta d_{min}$ , то между рассматриваемыми векторами вставляется новый вектор. Вставляемому вектору присваивается маркер “Препятствие”, а его начальной и конечной точке присваиваются маркеры “Угол выступающий” или “Угол утопленный” в соответствии с маркерами точек, лежащих под отрезком. Для реалий городской квартиры значение константы  $\Delta d_{min}$  принято равным 3 см.

Так как фильтрация исходной информации не устраняет полностью проблему шумов, алгоритмы обнаружения скачков и углов, как правило, генерируют некоторое количество ложных углов. Для того чтобы избавиться от ложных углов, используется алгоритм слияния векторов. В этом алгоритме предполагается, если два соседних вектора формируют угол, близкий к  $180^\circ$ , то этот угол ложный и векторы надо слить в один вектор. Кроме того, каждая пара векторов исследуется на возможность их слияния по сумме их длин. Слияние (рис. 11) считается возможным, когда

$$\mathbf{L}_m + \mathbf{L}_k - \mathbf{L}_{mk} < \Delta \mathbf{L}_{max},$$

где:

$\mathbf{L}_m$  – длина вектора  $m$ ,

$\mathbf{L}_k$  – длина вектора  $k$ ,

$\mathbf{L}_{mk}$  – расстояние от начала вектора  $m$  до конца вектора  $k$ ,

$\Delta \mathbf{L}_{max}$  – некоторая пороговая величина, которая подбирается эмпирически и принята равной 1.5 см.

$k=m+1$ , т.е. векторы – соседние.



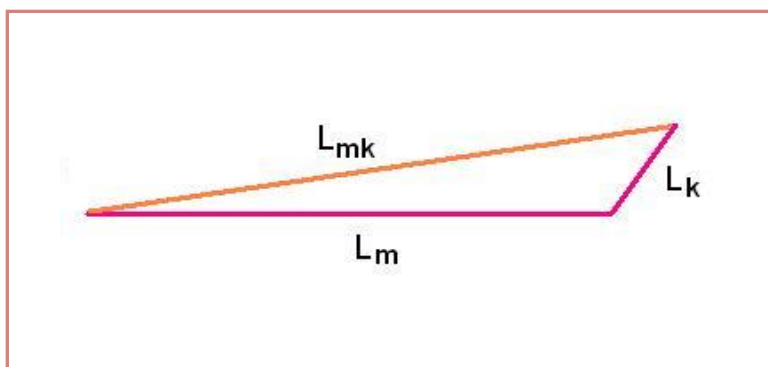


Рис.11. Слияние отрезков

Слияние допустимо только для векторов с одинаковыми маркерами. При слиянии двух векторов маркер и координаты конца/начала длинного вектора замещаются маркером и координатами начала/конца короткого вектора, после чего короткий вектор удаляется из массива.

В результате работы алгоритма уничтожения разрывов весь сектор обзора сканера оказывается закрытым массивом векторов. Этот массив векторов должен ограничивать область пространства перед роботом, по которой, на данном этапе, робот можно передвигаться, не рискуя столкнуться с препятствиями. Однако в этой области все еще могут существовать зоны, недоступные для робота. Это зоны, расположенные между выступающими углами и являющиеся такими узкими проходами, что робот не может войти в них.

Поэтому на следующем этапе алгоритма в полученный массив границ дальномерной видимости вставляются векторы, закрывающие узкие проходы между выступающими углами. Эти векторы получают новый маркер **“Прохода нет”** (см. рис.12). Меняется последовательность векторов в массиве, а часть векторов, окаймляющих зону с узким проходом, удаляется.

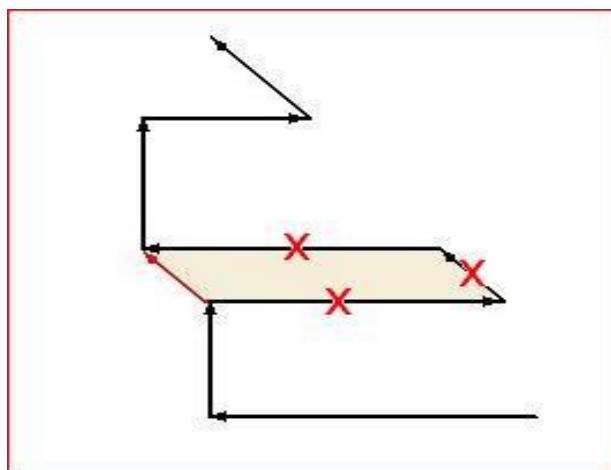


Рис. 12. Вставка отрезка “Прохода нет”

### 2.3. Построение массива Пассфреймов

Массив семантически помеченных векторов ограничивает область пространства перед роботом, по которой он, теоретически, может передвигаться, не рискуя столкнуться с препятствиями. Однако область эта не обязательно выпуклая, что затрудняет построение безопасного движения робота по траектории, которая гарантировала бы правильность обхода препятствий без захода в тупики. Для упрощения процедуры построения маршрута движения область пространства перед роботом разбивается на отдельные зоны – выпуклые многоугольники, отделенные друг от друга векторами, которые робот может пересекать.

С этой целью последовательно анализируется массив векторов, построенный на предыдущем этапе, и находятся все соседние пары выступающих углов, между которыми есть хотя бы один утопленный угол. Каждая пара точек, соответствующая двум выступающим углам, соединяется вектором, которому присваиваются маркеры **“Проход”**, а полученный таким образом выпуклый многоугольник (см. рис. 13) заносится в новый массив – массив Пассфреймов. Векторы, попавшие в сформированный Пассфрейм, заменяются в массиве векторов одним новым вектором, который по длине равен, но по направлению противоположен сформированному вектору **“Проход”** в Пассфрейме. Далее процедура повторяется до тех пор, пока сам массив векторов не превратится в выпуклый многоугольник.

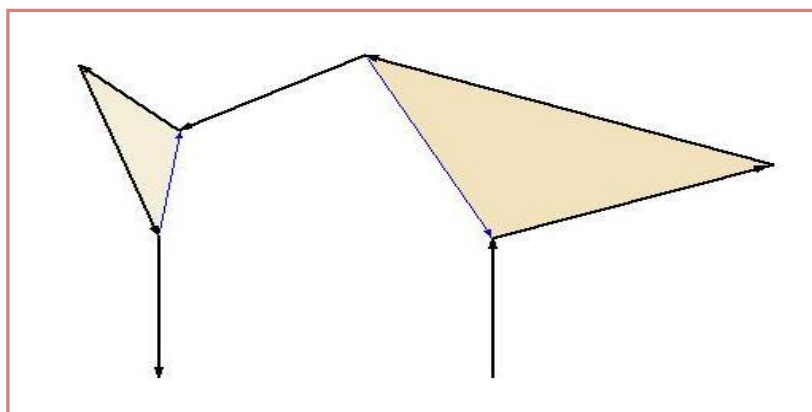


Рис.13. Выделение Пассфреймов из массива отрезков

Всем Пассфреймам присваивается один из следующих семантических маркеров-типов:

- “Тупик”;
- “Проход”;
- “Возможен проход”;
- “Граница видимости”;
- “Закрытая зона ”.

Пассфрейм получает маркер **“Закрытая зона”**, если все его векторы помечены маркерами **“Прохода нет”** или **“Препятствие”**. Пассфрейм получает маркер **“Тупик”**, если все его векторы помечены маркерами **“Прохода нет”** или **“Препятствие”**, кроме одного, который помечен маркером **“Проход”**. Пассфрейм получает маркер **“Проход”**, если как минимум два его вектора помечены маркером **“Проход”** и нет векторов типа **“Граница видимости”**, **“Скачек удаление”** или **“Скачек приближение”**.

Пассфрейм получает маркер **“Возможен проход”**, если хотя бы один из его векторов помечен маркером **“Скачек удаление”** или **“Скачек приближение”**, есть хотя бы один вектор **“Проход”** и нет векторов типа **“Граница видимости”**. Пассфрейм получает маркер **“Граница видимости”**, если хотя бы один из его векторов помечен маркером **“Граница видимости”** и есть хотя бы один вектор **“Проход”**.

Два соседних Пассфрейма можно слить в один, если в результате получится также выпуклый многоугольник. Алгоритм слияния Пассфреймов очевиден. Два вектора, разделяющие Пассфреймы, удаляются, а остальные перестраиваются в порядке обхода области внутри нового, объединенного Пассфрейма против часовой стрелки.

#### ***2.4. Характеристики программного обеспечения***

Описанный в данном разделе алгоритм реализован как часть программного обеспечения системы управления мобильного робота РБ-2. Эта программа используется и для формирования Карты операционной среды, и для построения Оперативной модели в процессе движения робота. Программное обеспечение системы управления работает циклически, в течение интервала времени между получением двух массивов данных от лазерного сканера. Продолжительность цикла сканирования составляет 100 мсек. На каждом цикле работы сканера оперативная модель полностью перестраивается, формируется маршрут и координатная траектория движения. В течение того же интервала в 100 мсек три раза вырабатываются управляющие сигналы для контроллера приводов робота. Кроме того, система управления обеспечивает также функционирование всех остальных подсистем робота. Естественно, в этих условиях быстроедействие алгоритма является одним из основных его характеристик.

Программа системы управления роботом испытывалась в среде обычной городской квартиры. В ходе испытаний оценивалось и быстроедействие программного обеспечения. Результаты испытаний показали, что время на формирование одной Оперативной модели, состоящей примерно из 15-25 Пассфреймов, не превышает 15 мсек. Такая скорость позволяет не только управлять движением робота в режиме реального времени, но и в интервале цикла сканирования оставляет большой запас времени для всех остальных программных процессов, выполняемых на компьютере робота.

Построенная Статическая модель среды хранится в базе данных робота, откуда запрашивается по мере необходимости. Она используется при планировании и при корректировке маршрута движения робота. В этих условиях удобными характеристиками описываемой модели являются её структурированность и компактность, которые позволяют быстро находить и извлекать необходимую информацию из базы данных.

Испытания показали, что в среде городской квартиры комната площадью в 15 кв. м. описывается не более чем 25-35 Пассфреймами, а процедуры реакций на запросы к базе данных, построенной даже на такой медленной платформе, как Access, не превышают 5 мсек.

### 3. Построение маршрута и траектории движения робота

Массив Пассфреймов – это одновременно и координатный, и топологический образ свободного от препятствий пространства вокруг робота, по которому он может передвигаться. В системе управления робота РБ-2 этот массив используется для построения маршрута движения, который, в свою очередь, является источником информации для формирования и реализации координатной траектории движения.

Построение маршрута начинается с того, что идентифицируется Пассфрейм, в контуре многоугольника которого находится робот (точнее, лазерный сканер, установленный на роботе). Для этого ищется Пассфрейм, для которого точка положения сканера является внутренней точкой. Этот Пассфрейм назначается исходным узлом. С него начинается поиск в сети цепочки узлов к целевому Пассфрейму.

Выбор целевого узла сети зависит от задачи движения мобильного робота:

Если нужно как можно ближе подойти к находящейся в помещении к точке с координатами  $(X, Y)$ , то выбирается Пассфрейм с любым маркером, кроме маркера **“Закрытая зона”**, для которого точка с координатами  $(X, Y)$  является внутренней. Если такого Пассфрейма нет, то выбирается Пассфрейм с маркером **“Граница видимости”** или **“Возможен проход”**, который находится ближе всего к точке  $(X, Y)$ .

Если нужно как можно дальше пройти по помещению вдоль заданного направления, то выбирается самый удаленный Пассфрейм с маркером **“Граница видимости”** или **“Возможен проход”**, который ближе всего к заданному направлению.

Если же нужно обойти все помещение, в котором находится робот, то выбирается самый близкий к роботу Пассфрейм с маркером **“Граница видимости”** или **“Возможен проход”**.

Во всех этих случаях алгоритм построения маршрута движения сводится к задаче формирования цепочки Пассфреймов. Решается она методом прямого

перебора узлов сети в глубину с возвратами и с проверкой на цикличность. Для каждого уже выбранного узла сети выбирается соседний узел – Пассфрейм, который с предыдущим Пассфреймом имеет общую границу, помеченную маркером “**Проход**”. Идентификатор вектора - границы Пассфрейма формирует очередную подцель маршрута движения. Процесс перебора начинается от исходного узла и завершается, когда на очередном шаге будет выбран целевой узел. Результатом работы алгоритма являются упорядоченные цепочки Пассфреймов, причем каждая цепочка представляет собой один из возможных маршрутов движения по сети из исходного узла в целевой узел. В реалиях обычной городской квартиры, как правило, существует один единственный маршрут, в редких случаях – два. Среди этих маршрутов сейчас выбирается тот, который удовлетворяет критерию минимальной длины. Очевидно, что могут быть и другие критерии выбора удобного маршрута.

Координатная траектория движения строится из отрезков прямых, проходящих через все Пассфреймы в цепочке выбранного маршрута. Эти отрезки соединяют последовательно, от предыдущего Пассфрейма к следующему, координаты *средних точек* векторов с маркером “**Проход**”, выбранных в качестве подцелей маршрута движения.

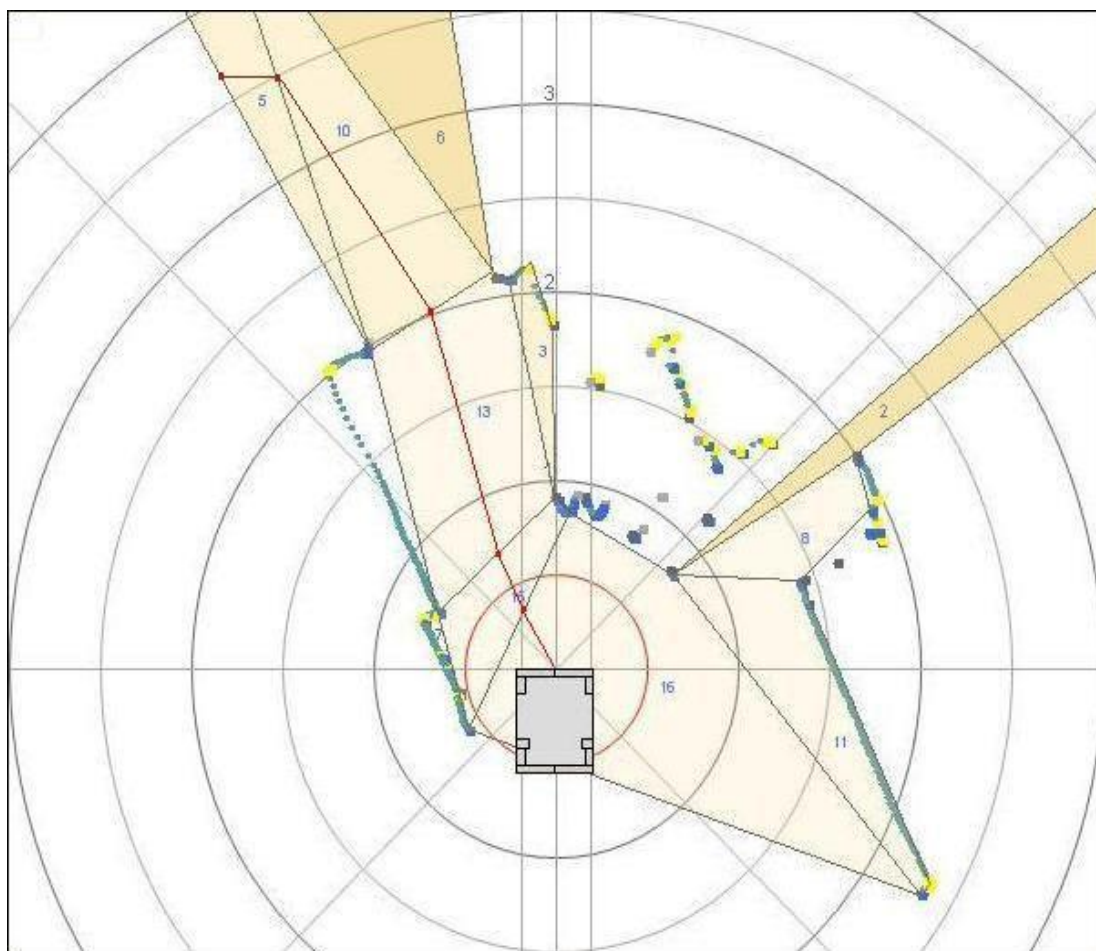


Рис. 14. Построение траектории движения (пример реальных измерений)

На основе описанного выше алгоритма создана программа, которая является частью системы управления робота РБ-2. На рис. 14 показан результат работы этой программы вместе с программой формирования сети Пассфреймов. Вычисления проводились на реальных данных, полученных с лазерного сканера на роботе, находящемся в среде реальной городской квартиры. Хорошо видна геометрия поверхности предметов, окружающих робота. Слева, перед роботом, и справа, за роботом, видны поверхности шкафов, стоящих вдоль стен комнаты. Справа перед роботом видны четыре ножки стула. За стулом – тумба письменного стола и свободное пространство под столом. Рядом со стулом, чуть дальше от робота, отчетливо видны две ноги человека, стоящего рядом со стулом.

Пассфреймы как бы огибают все препятствия, оставляя перед роботом пространство, по которому он может двигаться. На рисунке 15 приведена топология сети Пассфреймов, сформированной программным обеспечением робота. Желтым цветом окрашены узлы, которые выбраны как маршрут движения. Координатная траектория движения, построенная в соответствии с этим маршрутом, изображена на рисунке 14 ломаной линией красного цвета. Она идет от робота к двери, находящейся между Пассфреймом 13 и Пассфреймом 10, и далее, от двери – в смежное помещение.

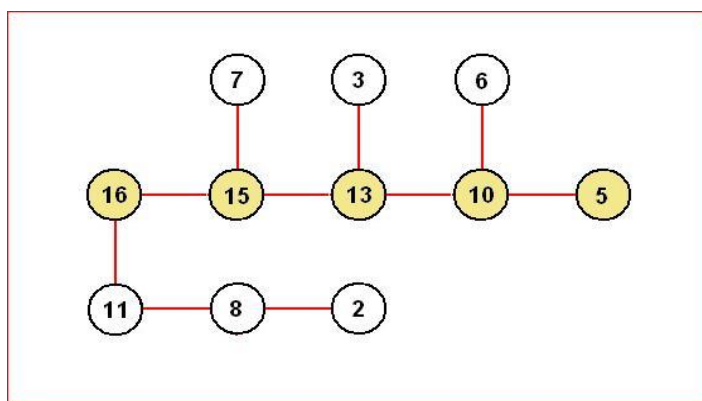


Рис. 15. Топология сети Пассфреймов

Заметим, что двери заведомо являются источниками изменений состояния операционной среды робота. Вместе с этим, факты наличия или отсутствия дуги 13-10 в построенной сетевой модели и векторные параметры этой дуги на логическом уровне отвечают на вопрос: открыта ли, закрыта ли дверь, а на координатном уровне модели среды: как именно она полуоткрыта?

Здесь легко увидеть присущее модели сети Пассфреймов очень важное свойство наблюдаемости любых изменений окружения робота.

Испытания программы построения маршрута и координатной траектории движения робота показали, что на сети, состоящей из 15 Пассфреймов, время работы программы для построения траектории длиной 500 см не занимает более 1 мсек. Этот факт позволяет наблюдать и учитывать достаточно быстрые изменения среды движения робота.

#### 4. Формирование и использование семантической модели

Рамки данной публикации не позволяют привести полное описание алгоритма работы логическо-декларативного уровня системы управления. Поэтому в данном и в последующем разделе представлены только основные принципы его устройства и приводятся примеры, дающие общее представление об использовании сети Пассфреймов на уровне планирования поведения робота.

На логическо-декларативном уровне системе управления робота РБ-2 для построения логических умозаключений реализован классический алгоритм Графплан, описанный в [15]. В качестве языка декларативного описания используется синтаксически видоизмененная русифицированная версия языка STRIPS [16]. Несмотря на критику этого языка в [17], на ограниченность его выразительных возможностей и на наличие скрытых недостатков, язык, тем не менее, удобен для использования на логическо-декларативном уровне системы управления роботом и хорошо приспособлен для работы и с Пассфреймами. Структура Пассфреймов и наличие в них семантических маркеров делают предельно простым алгоритм извлечения соответствующих предикатных выражений на языке STRIPS. В основу языка заложены лишь несколько типов предикатных выражений, приспособленных именно к задачам логики выбора.

Ниже приводится пример декларативного описания операционной среды, сгенерированный системой управления робота РБ-2 на сети Пассфреймов, описанной в предыдущем разделе. В этом примере используются только два типа предикатов: "Объект" и "Факт". Как сам пример, так и алгоритм извлечения предикатов из Пассфреймов просты и интуитивно понятны.

**(Объекты:**

**Пассфрейм(ПФ16)**

**Пассфрейм(ПФ15)**

**Пассфрейм(ПФ13)**

**Пассфрейм(ПФ11)**

**Пассфрейм(ПФ10)**

**Пассфрейм(ПФ8)**

**Пассфрейм(ПФ7)**

**Пассфрейм(ПФ6)**

**Пассфрейм(ПФ5)**

**Пассфрейм(ПФ3)**

**Пассфрейм(ПФ2)**

)

(Факты:

**Проход-из-в(ПФ16,ПФ15)**

**Проход-из-в(ПФ16,ПФ11)**

**Проход-из-в(ПФ11,ПФ8)**

**Проход-из-в(ПФ8,ПФ2)**

**Проход-из-в(ПФ15,ПФ7)**

**Проход-из-в(ПФ15,ПФ13)**

**Проход-из-в(ПФ13,ПФ3)**

**Проход-из-в(ПФ13,ПФ10)**

**Проход-из-в(ПФ10,ПФ6)**

**Проход-из-в(ПФ10,ПФ5)**

)

Добавим к фактам описание положения исходного Пассфрейма:

**Робот-находится-в(ПФ16)**

Зададим в качестве целевого предиката:

**Робот-находится-в(ПФ5)**

Зададим теперь и единственное декларативное правило поведения:

(Действие: Движение-из-в

(Параметры:

**Пассфрейм(?x)**

**Пассфрейм(?y)**

)

(Предусловие:

**Робот-находится-в(Пассфрейм(?x))**

**Проход-из-в(Пассфрейм(?x),Пассфрейм(?y))**

)

(Эффект+:

**Робот-находится-в(Пассфрейм(?y))**

)

(Эффект-:

**Робот-находится-в(Пассфрейм(?x))**

)

)

Предложенные выше предикатные выражения являются исходной информацией для алгоритма Графплан. Результатом работы программы является последовательность предикатов, описывающая маршрут движения робота:

**Робот-находится-в(ПФ16)**

**Робот-находится-в(ПФ15)**

**Робот-находится-в(ПФ13)**

**Робот-находится-в(ПФ10)**

**Робот-находится-в(ПФ5)**



## 5. Расширение семантической модели

В разделе 1.1 настоящей работы указывалось, что для решения задачи построения маршрута движения топологические модели более эффективны. Описанный в предыдущем разделе пример семантической модели является лишь демонстрацией возможности использования Пассфреймов для построения логических умозаключений и принятия решений. Подсистема логическо-декларативного уровня системы управления роботом РБ-2 не решает задач построения маршрутов. Эти задачи решаются на перцепторно-моторном уровне алгоритмов управления движением. Задача подсистемы логическо-декларативного уровня – формировать логику поведения робота.

Логическо-декларативный уровень системы управления имеет обобщенную модель операционной среды, состоящую из деклараций объектов, фактов и предположений. Эта модель в настоящее время формируется человеком-оператором и хранится в базе данных системы управления робота. Кроме того, в этой же базе данных хранится набор правил, также сформированный человеком, используемых для рассматриваемого типа операционной среды.

Объектами и их параметрами, которыми оперирует логическо-декларативный уровень системы управления, являются, например:

**Помещение, Комната, Дверь, Окно, Проход, Мебель, Стол, Стул, Шкаф, Открыт, Закрыт.**

Факты, которые относятся к операционной среде, могут быть общими, определяющими отношения между типами и параметрами объектов, например:

**Это-есть(Комната, Помещение)**

**Это-есть(Стол, Мебель)**

**Это-есть(Стул, Мебель)**

**Это-есть(Шкаф, Мебель)**

**Это-не(Открыт, Закрыт)**

Факты могут быть и конкретными, определяющими отношения между конкретными объектами, например:

**Находится-в(Окно1, Комната1)**

**Находится-между(Проход12, Комната1, Комната2)**

**Находится-между(Дверь23, Комната2, Комната3)**

Среди объектов, используемых логическо-декларативной подсистемой, есть и Пассфрейм. Параметрами этого объекта являются:

- **Тип(маркер)** – определяется значением семантического маркера, присвоенного Пассфрейму в процессе создания;
- **Находится-в(Пассфрейм, Помещение)** – факт как параметр, определяющий идентификатор объекта, помещения, в котором находится Пассфрейм.

Кроме того, среди объектов, которыми оперирует логическо-декларативной уровень системы управления, есть и вектор-сторона многоугольника Пассфрейма, параметрами которого являются:

- **Тип(маркер)** – определяется значением семантического маркера, присвоенного вектору в процессе создания;
- **Граница-с(Пассфрейм,Вектор,Объект)** – факт, как параметр, определяющий идентификатор объекта, с которым граничит данный вектор Пассфрейма.

Пример:

**Находится-в(ПФ17, Комната1)**

**Граница-с(ПФ17,В5,Книжный-шкаф)**

**Граница-с(ПФ17,В7,Дверь)**

Это означает, что Пассфрейм с идентификатором **ПФ17** находится в границах помещения **Комната1**. По вектору **В5** он граничит с книжным шкафом, а по вектору **В7** - с дверью. Очевидно, что вектор **В5** должен иметь маркер “**Препятствие**”, а вектор **В7** - “**Проход**”.

Семантические маркеры, определяющие типы векторов и типы Пассфреймов, заносятся в структуру Пассфреймов автоматически, в процессе их создания. А вот значения параметров-фактов, определяющих положение Пассфрейма относительно предметов помещения, а также определяющих имена объектов, с которыми граничат векторы Пассфрейма, программным путем сформировать невозможно. Это обусловлено тем, что в описании среды, с которым оперирует логическо-декларативный уровень, присутствуют только имена таких объектов и полностью отсутствуют их координаты.

Именно наличие параметров-фактов, определяющих как имя объекта, в котором находится сам Пассфрейм, так и имена объектов, окружающих Пассфрейм, обеспечивает нужную связь между чисто декларативной моделью логическо-декларативного уровня управления и комбинированной моделью сети Пассфреймов перцепторно-моторного уровня.

На данном этапе присваивать значения этим параметрам может только человек-оператор. Значения этих параметров записываются в Базу Данных вручную. Это входит в процедуру построения Статической модели среды и делается с помощью стандартного интерфейса платформы Access сразу после занесения Статической модели в базу данных. В дальнейшем будет создан интерфейс для присваивания значения этим параметрам в форме диалога.

## 6. Заключение

Основным результатом этой работы является предложенная и реализованная модель операционной среды мобильного робота. Модель представляет собой комбинацию метрической, топологической и семантической моделей, параметры которых хранятся в специальной базе данных. Основным объектом модели является Пассфрейм - свободный от препятствий выпуклый многоугольник на горизонтальной плоскости, по которой движется робот. Сам многоугольник и ограничивающие его векторы помечены семантическими маркерами. Пассфреймы объединяются в сеть, дуги которой описывают достижимость Пассфреймов. Все эти операционные сведения составляют содержание таблиц реализованной базы данных системы управления мобильного робота РБ-2 вместе с перечнем возможных запросов, формируемых программными модулями системы управления.

В работе разработан и подробно описан эффективный алгоритм автоматического формирования сети Пассфреймов на данных, получаемых с лазерного сканера. На базе этого алгоритма создана программа для мобильного робота РБ-2, которая успешно отлажена и испытана. Программа формирует модель операционной среды, состоящую в среднем из 15-20 Пассфреймов, за время, не превышающее 15 мсек.

Испытаны алгоритмы формирования топологического маршрута и координатной траектории мобильного робота на базе сети Пассфреймов. Программа, созданная на основе этих алгоритмов, относится к перцепторно-моторному уровню системы управления мобильного робота РБ-2. Программа формирует маршрут и координатную траекторию на расстояние около 5 м меньше чем за 1 мсек.

В работе на простых примерах продемонстрирована возможность использования сети Пассфреймов также и в качестве семантической модели операционной среды на верхних уровнях управления двигательным поведением робота. Это позволяет продолжить данную работу, направив ее на развитие алгоритмов логическо-декларативного уровня системы управления мобильного робота РБ-2 и структуры его базы данных.

## Список литературы

1. Richard E. Fikes, Nils J. Nilsson (Winter 1971). "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving". *Artificial Intelligence* 2 (3–4): 189–208.
2. Охоцимский Д.Е., Платонов А.К. Алгоритмы управления шагающим аппаратом, способным преодолевать препятствия. Изв. АН СССР. Техн. кибернетика, 1973, №5. с. 3-14.
3. Охоцимский Д.Е., Платонов А.К. Перцептивный робот, перемещающийся в трёхмерной среде. Труды IV Международной объединённой конференции по искусственному интеллекту. Том 9. М.: Наука. 1975. с. 9.161
4. Кирильченко А.А. О представлении информационно-двигательного взаимодействия мобильного робота со средой на основе отношения видимости. М.: Препр. ИПМ им. М.В. Келдыша АН СССР, 1987, № 235, 28 с.
5. J. B. Oommen, S. S. Lyengar, N. S. V. Rao, R. L. Kashyap, Robot navigation in unknown terrains using learned visibility graphs. *IEEE Robotics Automation*, vol. RA-3, no. 6, Dec. 1987.
6. Кирильченко А.А. Об исследовании эффективности алгоритмов выбора пути в условиях неопределенности. 2. Атлас особых ситуаций и атлас "неустойчивого доминирования". М.: Препр. ИПМ им. М.В.Келдыша РАН, 1997, № 44. 27 с.
7. H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, S. Thrun. *Principles of Robot Motion - Theory, Algorithms, and Implementation*. The MIT Press, 2005.
8. Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
9. H. Choset, K. Nagatani. Topological simultaneous localization and mapping (T-SLAM). *IEEE Transactions on Robotics Automation*, 17, April 2001.
10. J.-S. Gutmann, T. Weigel, B. Nebel. A fast, accurate, and robust method for self-localization in polygonal environments using laser-range-finders. *Advanced Robotics Journal*, 14(8), 2001.
11. R. Siegwart, I. R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. The MIT Press, 2004.
12. S. Thrun, A. Buckenz. Integrating Grid-Based and Topological Maps for Mobile Robot Navigation. *Proceedings of the Thirteenth National Conference on Artificial Intelligence AAAI*, Portland, Oregon, August 1996.
13. D. Filliata, J.-A. Meyerb. Map-based navigation in mobile robots. *Cognitive Systems Research*, 4, 2003.
14. B. Weiss. Fast Median and Bilateral Filtering. *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, 2006.
15. С.Рассел, П.Норвиг. Искусственный интеллект: Современный подход. / Пер. с англ., М.: Издательский дом «Вильямс», 2006.
16. Fikes R. E. and Nilsson N. J. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4), 1971.
17. V. Lifschitz. *On the Semantics of STRIPS. Reasoning about Actions and Plans*. Morgan Kaufmann Publishers, Inc., 1987.

## Оглавление

<b>1. Введение .....</b>	<b>3</b>
1.1. Основные подходы к созданию модели операционной среды .....	3
1.2. Мобильный робот РБ-2 .....	5
1.3. Свойства модели операционной среды робота РБ-2 .....	7
<b>2. Формирование модели операционной среды .....</b>	<b>8</b>
2.1. Построение массива семантически помеченных точек .....	11
Алгоритм обнаружения скачков дальности .....	11
Алгоритм обнаружения углов-изломов .....	12
2.2. Построение массива семантически помеченных векторов .....	13
2.3. Построение массива Пассфреймов .....	17
2.4. Характеристики программного обеспечения .....	18
<b>3. Построение маршрута и траектории движения робота .....</b>	<b>19</b>
<b>4. Формирование и использование семантической модели .....</b>	<b>22</b>
<b>5. Расширение семантической модели.....</b>	<b>24</b>
<b>6. Заключение .....</b>	<b>26</b>
<b>Список литературы.....</b>	<b>27</b>