

# Эвристический алгоритм обнаружения изолированных препятствий мобильным роботом по дальномерным данным<sup>1</sup>

**Аннотация.** Описывается алгоритм, решающий задачу обнаружения одиночных изолированных препятствий мобильным роботом с помощью дальномера. Основной блок алгоритма построен как система продукционных правил, которые вводят логические соотношения, позволяющие судить о наличии или отсутствии препятствия по полю нормалей к поверхности. Найденные препятствия заносятся на двумерную карту. Обсуждаются методы картирования препятствий.

**Ключевые слова:** мобильный робот, дальномер, продукционная система.

## Введение

Лазерные дальномерные сканирующие сенсоры, иначе называемые лидарами [1, 2], в настоящее время широко применяются в мобильной робототехнике. При этом решаются следующие наиболее типичные задачи [3-9]:

- реализация системы безопасности (бесконтактного бампера);
- обнаружение и картирование препятствий (при известной навигационной информации);
- локализация мобильного аппарата и составление карты, SLAM-задачи;
- контроль профиля дороги;
- совместное использование данных (data fusion) дальномеров и других сенсоров, например, ТВ-камер, тепловизоров, при этом точнее выделяются объекты наблюдаемой сцены и определяются расстояния до них.

Внешний вид одной из моделей, дальномера SICK LMS 200, приведен на Рис. 1 (с ним проводились авторские эксперименты). Основные характеристики прибора: измеряемая дальность до

50 м, горизонтальная развертка 180 град. с шагом измерительных лучей в 1 град., точность измерений на дальностях более 5-10 м порядка 1 см.

Некоторым недостатком этого прибора является его вес 4.5 кг, что не позволяет использовать дальномер на малых роботах. Однако в настоящее время появились небольшие лидары, так что этот недостаток уже преодолен различными фирмами.

На Рис. 2 показаны участники соревнований Darpa Challenge (гонки беспилотных автомобилей в пустыне, США). Слева показан автомобиль-робот с дальномерами SICK LMS 291 (команда ENSCO TEAM, Университет штата Вирджиния), справа – автомобиль команды Tartan Racing Университета Carnegie-Mellon (Питтсбург, штат Пенсильвания) [9]. На автомобиле слева установлено на переднем бампере 3 дальмера SICK, один из них «смотрит» вниз для контроля профиля дороги, на автомобиле справа можно видеть 5 дальномеров, из которых два установлены на бампере, три – на крыше. Ясно, что такая сенсорная оснащенность позволяет машинам эффективно контролировать дорогу.

<sup>1</sup> Работа выполнена при поддержке РФФИ, проекты 16-08-00880-А, 16-01-00131-А, 15-07-07483-А.

Подготовлена как расширенный вариант доклада, представленного на Конференцию по Искусственному Интеллекту КИИ-2016. Смоленск, 3-7.10.2016.



Рис. 1. Лазерный дальномер SICK LMS 200 (Германия)



Рис. 2. Автомобили – участники соревнований Darpa Challenge (гонки в пустыне), оснащенные дальномерами SICK

На Рис. 3 показан российский беспилотный автомобиль КАМАЗ, разрабатываемый КАМАЗом и российской компанией Cognitive Technologies [8]. На фотографии виден дальномер на бампере машины. Такое его расположение позволяет контролировать препятствия, появляющиеся впереди машины.

Важным элементом в системе управления мобильного робота является алгоритм, решающий задачу обнаружения им одиночных изолированных препятствий во время передвижения в естественной среде. Покажем, что для этого алгоритма эффективным сенсором служит дальномер. Во время сканирования дальномер определяет точки пересечения его линии визирования (наблюдения) с окружающими робот объектами - опорной поверхностью и препятствиями. Однако с «точки зрения» робота эти точки первоначально являются равнозначными. Задача состоит в том (ставится так), чтобы среди всего этого множества (облака) точек выделить точки, принадлежащие опорной поверхности, и точки, принадлежащие препятствиям, определить размеры этих препятствий и нанести их на 2D-карту местности.



Рис. 3. Беспилотный КАМАЗ

Будем считать, что изучаемая местность представляет собой достаточно ровную поверхность с изолированными препятствиями (например, они могут представлять собой одиночные камни или груды камней). Будем также считать, что из показаний других приборов (помимо дальномера) роботу известно направление местной вертикали.

## 1. Исходные определения

Поставленная задача будет решаться для автомобиля-робота проекта АвтоНИВА [10-12], разработка которого ведется в ИПМ им. М.В. Келдыша при участии РГГУ, при поддержке РФФИ. На Рис. 4, а показана машина и модель ее сенсорной информационной системы (Рис. 4, б), которая будет установлена на крышу. Основным ее элементом для рассматриваемой задачи является дальномер SICK.

Будем считать с достаточной общностью, что участки поверхности являются проходимыми для робота, если нормали в точках поверхности такого участка мало отличаются от вертикали (Рис. 5, а). Так будет, например, если местность представляет собой пологие «холмы». Малое отличие от вертикали главным образом зависит от наклона поверхности данного участка по отношению к горизонтальной плоскости  $Oxy$  и определяется конструктивными параметрами робота. Будем считать, что участок наклонной поверхности является участком подстилающей поверхности, если робот может, как спуститься, так и подняться по такому

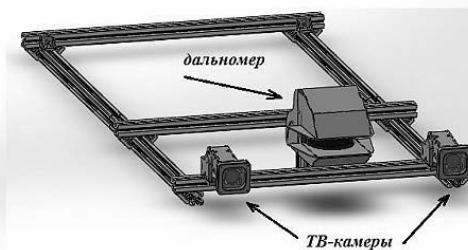
участку. Пример такой поверхности приведен на Рис. 5, б. Закрашены непроходимые участки с большими углами наклона к горизонтали  $c\theta$ .

Будем считать, что у поверхности нет значительных локальных перепадов высоты в исследуемом участке, т.е. поверхность не представляет собой «ступеньки». Предположим, что на подстилающей поверхности расположено одиночное препятствие (некоторое тело), так, как показано на Рис. 5, б. Рассмотрим изменение направлений нормалей в плоскости  $P$ , следуя за подъемом луча дальномера, исходящего из точки  $D$ .

Нормаль в точках, принадлежащих опорной поверхности, вначале будет мало отличаться от вертикали. В точках, принадлежащих телу, направление нормали значительно изменяется. Сначала нормаль сильно отклоняется от вертикали и направлена в полупространство I, определяемое точкой подвеса дальномера  $D$  и плоскостью  $R$ , перпендикулярной плоскости  $P$  (т.е. нормаль противоположна лучу визирования, скалярного произведения их направляющих векторов отрицательно). Затем нормаль снова становится почти вертикальной, а затем направляется в полупространство II, являющееся

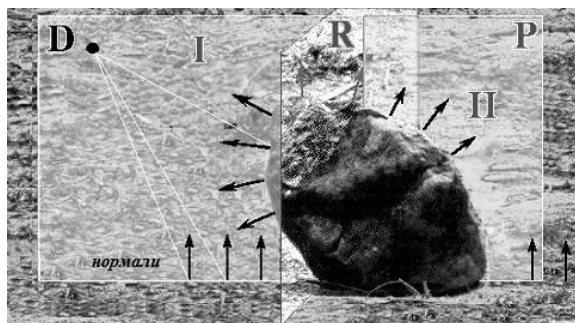


а)

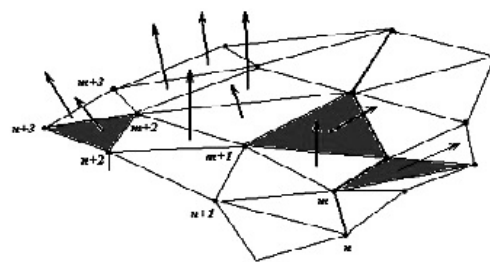


б)

Рис.4. Автомобиль-робот проекта АвтоНИВА (а), его сенсорная информационная система-модель (б)



а)



б)

Рис. 5. Классификация зон проходимости по нормальям

дополнением полупространства I (нормаль со-направлена с лучом визирования, скалярного произведения их направляющих векторов положительно). Затем снова в точках, принадлежащих опорной поверхности, нормаль будет мало отличаться от вертикали. Изменение направлений нормалей показано на Рис. 5, а. Эвристическая схема предлагаемого алгоритма заключается в слежении за указанным «поведением» нормалей к опорной поверхности.

## 2. Формальное определение метода обработки измерений

Примем, что результатом работы дальномерной системы является двухмерный скан - структура данных, в которой значения дальности являются функцией двух углов  $\alpha$  и  $\beta$  и определяются сферической системой координат, присущей дальномеру. Эта структура может быть произведена как реальным устройством, так и моделью дальнометрического сенсора, вычисляющей расстояния до сцены и имеющей тот же формат данных.

Введем метод обработки измерений дальномера, позволяющий построить 2D-карту препятствий. Этот метод включает в себя следующие 8 этапов:

- 1) вычисление трехмерных координат  $(x, y, z)$  всех измеренных точек по измерениям наклонных дальностей до них, множество этих точек обозначим  $P_n$ ;
- 2) построение триангуляционной сетки  $C_t$  (специальной сетки треугольников), натянутой на измеренные точки;
- 3) вычисление нормалей к построенным треугольникам сетки  $C_t$ ;
- 4) вычисление набора характеристических параметров  $H_s$  треугольников сетки  $C_t$  и построение векторов кодов достоверности  $C_d$  точек (вершин) треугольников  $C_t$ ;
- 5) анализ нормалей и векторов характеристических параметров  $H_s$  и кодов достоверности и выделение на этой основе зон проходимых участков, препятствий и их теней на 2D-карте;
- 6) деление зон препятствий на отдельные препятствия;
- 7) определение границ препятствий и уточнение информации о вершинах и треугольниках сетки  $C_t$ ;

8) построение карты и габаритов препятствий на 2D-карте, гарантированных габаритов, включающих и препятствия, и их тени.

Этапы 1-4 – подготовительные, этап 5 – основной, в нем с помощью логических правил принимается решение о наличии или отсутствии препятствий. На этапах 6-8 препятствие картируется – заносится на карту.

Заметим, общую информацию о дальномере можно найти в [1-4].

**Этап 1. Вычисление трехмерных координат измеренных точек.** Обозначим через  $d_n$  расстояние от точки D дальномера до  $n$ -й точки сцены,  $\alpha_n, \beta_n$  – значения измерительных углов линии визирования дальномера, соответствующих точке с индексом  $n$ . Тогда координаты  $n$ -ой точки (из облака) могут быть вычислены по формулам:

$$\begin{cases} x_n = d_n \cdot \cos \alpha_n \cdot \cos \beta_n \\ y_n = d_n \cdot \sin \alpha_n \cdot \cos \beta_n \\ z_n = z_D + d_n \cdot \sin \beta_n \end{cases}$$

Здесь  $z_D$  – высота подвеса дальномера над горизонтальной плоскостью  $Oxy$ . Заметим, что набор точек  $(x_n, y_n, z_n)$  в зарубежной литературе называется картой высот – *elevation map*.

**Этап 2. Построение триангуляционной сетки  $C_t$ .** На этом этапе выполняется специальная нумерация точек (вершин) и треугольников, построенных на этих точках. Результатом является триангуляционная сетка  $C_t$ , в дальнейшем все вычисления выполняются на этой сетке. Покажем эту нумерацию. Пусть  $i$  – номер очередного дискретного значения угла  $\alpha_i$  (в выходных данных дальномерной системы данные хранятся в порядке возрастания измерительного угла  $\alpha$ , которое выполняется против часовой стрелки со стороны положительного направления оси  $z$ ), а  $j$  – номер очередного дискретного значения угла  $\beta_j$ . Введем обозначение: пусть  $T(a, b, c)$  – треугольник, вершинами которого являются точки с номерами  $a, b, c$ . Построим треугольник  $T_k$ , соединив отрезками точки с номерами  $(iN_\beta + j, iN_\beta + j + 1, (i + 1)N_\beta + j)$  и треугольник  $T_{k+1}$ , соединив точки  $((i + 1)N_\beta + j, iN_\beta + j + 1, (i + 1)N_\beta + j + 1)$ .

В соответствии с введением индексов  $i, j, k$  для четных номеров  $k$  справедливо соотношение  $k = 2[(N_\beta - 1)(i - 1) + j - 1]$ , а для нечетных номеров  $k$  – соотношение:

$k = 2 \left[ (N_\beta - 1)(i - 1) + j - 1 \right] + 1$ , где  $N_t$  - число треугольников,  $N_t = 2(N_\alpha - 1)(N_\beta - 1)$ , треугольники  $T_k$  и  $T_{k+1}$  являются соседними. Все такие треугольники «организуют» измеренные точки в структуру сетки  $St$ .

**Этапы 3-4. Вычисление нормалей к треугольникам сетки  $St$ . Вычисление характеристических признаков и кодов точек.** Опишем вычисление нормалей к построенным треугольникам. Пусть имеется треугольник с номером  $k$  с достоверно измеренными вершинами. Коэффициенты уравнения плоскости  $A_k, B_k, C_k, D_k$ , проходящей через данный треугольник, могут быть рассчитаны по известным формулам геометрии, и вектор  $(A_k, B_k, C_k)$  – нормаль к треугольнику. Окончательно после нормировки можно получить вектор нормали единичной длины, направленный по  $(A_k, B_k, C_k)$ . Будем далее считать, что  $(A_k, B_k, C_k)$  – нормированный вектор. При соответствующем выборе номеров вершин нормали к треугольникам будут всегда направлены в одноименные полупространства (одного знака) относительно плоскостей этих треугольников, и если смотреть из конца вектора нормали к треугольнику, обход его вершин всегда будет осуществляться против часовой стрелки. Заметим, возможны и другие геометрические способы вычисления нормалей к треугольникам сетки  $St$ , но описанный – один из самых простых.

В рассматриваемых этапах вводится также специальная кодировка вершин – наборы характеристических признаков в системе. Это набор из направляющих косинусов, определяющих векторы системы, – нормали к  $k$ -му треугольнику триангуляционной сетки и линии визирования (измерительного луча) в связанной системе координат. Вводится также код, называемый «скачком высоты вперед».

**Этап 5. Выделение участков проходимости и зон препятствий и их теней на 2D-карте.** Введем вектор  $m_i$  маркировки треугольников сетки  $St$ . Этот вектор определяет раскраску сетки, при которой каждый треугольник с номером  $k$  может быть отнесен к одному из следующих типов: подстилающая поверхность ( $m_i(k)=0$ ), недостоверный треугольник ( $m_i(k)=-1$ ), препятствие ( $m_i(k)=-2$ ), боковая (относительно расположения пары дальномер-препятствие) тень от препятствия ( $m_i(k)=-3$ ) и тень от препятствия за

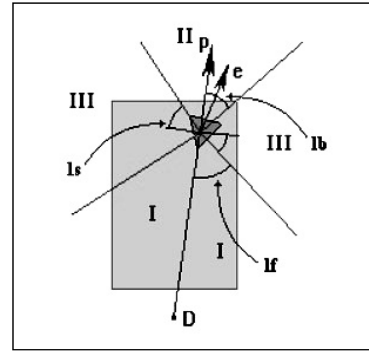


Рис. 6. Классификационные зоны вокруг препятствия

препятствием ( $m_i(k)=-4$ ). Опишем процедуру такой маркировки.

На Рис. 6 показаны проекция на плоскость  $D'x'y'$  (горизонтальная плоскость, проходящая через дальномер) некоторого тела (представляющего собой параллелепипед, в проекции – прямоугольник), проекции луча дальномера и нормали к выделенному треугольнику, а также передняя, задняя и две боковые классификационные зоны, обозначенные цифрами I, II и III соответственно. Введенные зоны реализуют пороговую классификацию по направлениям нормалей к исследуемым треугольникам.

Треугольники сетки  $St$  классифицируются по нескольким порогам, по ним же классифицируются вершины этой сетки:

$c0$  – порог угла наклона подстилающей поверхности;

$c1$  – порог угла наклона нормали к вертикали;

$c2$  – порог сонаправленности угла линии визирования с нормалью.

С помощью введенных пороговых величин будем далее классифицировать (раскрашивать) треугольники сетки  $St$  и их вершины. Для этого будем использовать два вектора признаков – треугольников  $m_i$  и аналогичный для вершин, значения которых последовательно модифицируются.

Введем набор решающих правил  $P = \{P1, P2, P3, P4, P5\}$ , выполнение определенного сочетания которых позволит отнести текущий треугольник  $T_k$  к одному из перечисленных выше типов. Правила  $Pi$  заключаются в выполнении следующих условий:

$P1$  – нормаль сильно отклонена от вертикали (угол наклона нормали больше порога  $c1$ );

$P2$  – проекция нормали на плоскость  $D'x'y'$  сонаправлена с проекцией луча обзора на ту же

плоскость - нормаль направлена в классификационную зону III (Рис. 6);

$P3$  - проекция нормали на плоскость  $D'x'y'$  противоположно направлена относительно проекции луча обзора на ту же плоскость - нормаль направлена в классификационную зону I;

$P4$  - в треугольнике нет скачка по  $z$ ;

$P5$  - треугольник с предыдущим номером определен как препятствие.

Тогда окончательно вывод о наличии/отсутствии препятствий может быть сделан следующими продукционными правилами:

если  $P_2 \wedge \overline{P_4}$  то  $m_i(k) = -4$ ,

если  $\overline{(P_2 \wedge \overline{P_4})} \wedge \overline{(P_2 \vee P_3 \vee P_4)}$   
то  $m_i(k) = -3$ ,

если  $\overline{\overline{(P_2 \wedge \overline{P_4})} \wedge \overline{(P_2 \vee P_3 \vee P_4)} \wedge (P1 \vee P3)}$   
то  $m_i(k) = -2$ .

Поясним эти условия. Если для треугольника  $T_k$  одновременно истинно правило  $P_2$  и ложно правило  $P_4$ , то такой треугольник маркируется (окрашивается) как тень за препятствием. Введенное условие означает, что нормаль к треугольнику направлена в классификационную зону II и в треугольнике имеется скачок высоты между его вершинами.

Если для треугольника  $T_k$  не является истинным логическое произведение  $P_2$  и  $\overline{P_4}$ , и если не является истинной логическая сумма правил  $P_2, P_3, P_4$  то такой треугольник маркируется как боковая тень. Это означает, что треугольник будет окрашен как боковая тень, если не выполнено предыдущее условие, и одновременно при этом нормаль к треугольнику не направлена в классификационные зоны I и II (т.е. направлена в зону III) и в треугольнике нет скачка высоты между его вершинами.

Если для треугольника  $T_k$  не является истинным предыдущее условие и если истинны условия  $P_1$  или  $P_5$ , то такой треугольник маркируется как препятствие. Это означает, что треугольник будет окрашен как препятствие, если одновременно не выполнено предыдущее составное условие, и либо нормаль сильно отклонена от вертикали (в зону I), либо предыдущий треугольник уже окрашен как препятствие.

**Этапы 6-7. Деление зон препятствий на отдельные препятствия. Определение границ препятствий.** На предыдущих этапах была

проведена маркировка вершин и треугольников сетки  $St$  и для каждой точки было установлено, принадлежит ли она какому-либо препятствию или нет. На завершающих этапах для каждой точки и для всех треугольников, маркированных как боковые тени или тени за препятствием, определяется, какому конкретному препятствию они принадлежат. Одновременно с этим строятся границы этих препятствий, проходящих через вершины треугольников. Границей будем называть множество тех вершин, у которых только часть содержащих их треугольников маркирована как препятствие.

**Этап 8. Построение карты. Построение габаритов и гарантированных габаритов препятствий и их теней на 2D-карте.** Здесь реализовано картирование препятствий. При отображении препятствий на карту представление информации о них должно удовлетворять следующим очевидным ограничениям.

Во-первых, информация о препятствиях должна занимать по возможности меньший объем в памяти компьютера. Во-вторых, она должна быть представлена так, чтобы обеспечить быстроедействие обработки данных. При этом зачастую не требуется детальной информации о препятствиях, важно лишь наличие данных об их характерных размерах и положении в пространстве (на 2D-карте). Примером этого являются задачи прокладки пути робота. Указанные данные могут быть представлены габаритами обнаруженных препятствий. Этап построения габаритов препятствий на карте - финишный этап алгоритма обнаружения препятствий.

После выполнения предыдущих этапов каждое препятствие будет выглядеть как облако точек, среди которых имеются внутренние и граничные точки препятствий. Дальнейшие построения, т.е. построения габаритов, выполняются на граничных точках

В описываемой системе, вообще говоря, не ставится задача построения минимального габарита. Поэтому алгоритм строит габариты нескольких типов (прямоугольники, окружности и эллипсы) по следующим причинам. Первый вариант является наиболее простым и быстрым, особенно если строить габаритные прямоугольники со сторонами, параллельными осям координат. Второй и третий важны тем, что дают области габаритов с гладкой границей. Это условие необходимо в дальнейшем для не-

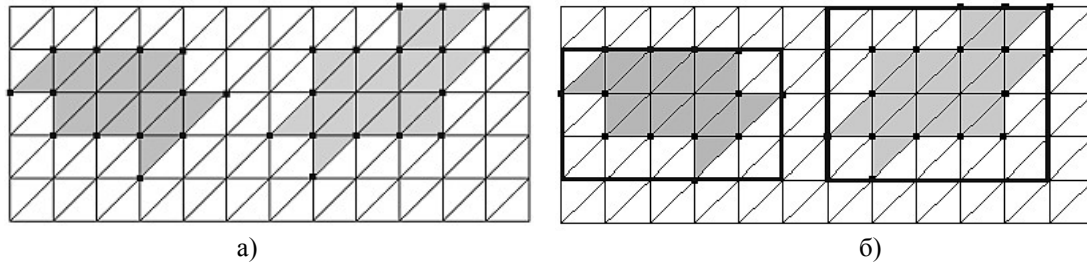


Рис. 7. Препятствия (а) и их габариты-прямоугольники (б)

которых алгоритмов прокладки пути робота, например, для метода потенциалов [13]. Для графовых методов прокладки маршрута, работающих на клетчатых сетках (сетках занятости), как например, волновой алгоритм, и другие методы прокладки пути [14], наоборот, удобнее габариты-прямоугольники. Всего в алгоритме реализованы 7 способов построения габаритов. Построен алгоритм так, что число способов нахождения габаритов может легко наращиваться.

**Алгоритм А1. Габарит - прямоугольник.** Рассмотрим построение прямоугольников, стороны которых параллельны осям координат. Такое построение является эффективным в случае препятствий, граничный диаметр (наибольшее расстояние между границами) которых не лежит вдоль биссектрис координатных квадрантов. Обозначим через  $RI$  множество

$$RI = \{RI_r\},$$

$$RI_r = \{MnX_r, MxX_r, MnY_r, MxY_r, 0 < r < No\} \quad (1)$$

- набор координат прямоугольников, соответствующих выделенным препятствиям, в нем заданы минимальные и максимальные значения их  $x$ - и  $y$ -координат. Последовательным перебором для всех точек границ препятствия алгоритм А1 определяет принадлежность их кортежу  $RI_r$ , т.е. находит точки с минимальными и максимальными значениями  $x$ - и  $y$ -координат. Фактически  $RI_r$  является габаритом-прямоугольником препятствия. Величины

$$D_r^x = MxX_r - MnX_r, D_r^y = MxY_r - MnY_r, \quad (2)$$

$$0 < r < No$$

- суть граничные диаметры построенных габаритных прямоугольников, т.е. их диагонали. Кортеж  $RI$  (1) и диаметры  $D_r^x$  и  $D_r^y$  (2) используются в А1 и также другими способами построения габаритов, описанными ниже.

Так. Алгоритм А1 получает набор прямоугольников, число которых равно числу препятствий, у которых минимальные  $x$ - и  $y$ -координаты совпадают с минимальными коор-

динатами точек границ препятствий, а максимальные  $x$ - и  $y$ -координаты совпадают с максимальными координатами точек границ препятствий. Пример построения габаритов в виде прямоугольников приведен на Рис. 7. Вычислительная сложность алгоритма такова - число его шагов есть однократный перебор вершин препятствий по границам препятствий.

**Алгоритм А2. Габарит - окружность.** Построение габаритных окружностей не единственно, в качестве габаритов-окружностей в этом алгоритме выбраны окружности, описанные вокруг прямоугольников (1), центр такой окружности совпадает с точкой пересечения диагоналей габаритного прямоугольника, а радиус равен половине его диагонали (2). Пример построения габаритов в виде окружностей общего вида приведен на Рис. 8, а.

Отметим, что так определенные окружности не являются минимальными, описанными вокруг облака точек окружностями, однако приведенное их вычисление отличается достаточной простотой и является вполне приемлемым для класса бортовых задач прокладки маршрута мобильного робота на местности.

**Алгоритм А3. Габарит - редуцированная (уменьшенная) окружность.** Опишем переборный алгоритм построения редуцированной габаритной окружности, он строит меньший габарит, нежели А2. Пример приведен на Рис. 8, б.

Алгоритм 3 линейно зависит от числа вершин препятствия и строит габаритную окружность итерационным процессом. Этот алгоритм имеет большую вычислительную сложность, чем А2, но позволяет в ряде случаев избежать недостатка предыдущего алгоритма, видимого на Рис. 8, где габаритные окружности пересекаются и превращают два препятствия в одно составное, хотя это и не является таковым.

Опишем алгоритм в виде последовательности шагов. Пусть точки  $P_r, 1 \leq r \leq N_i$  - вершины,

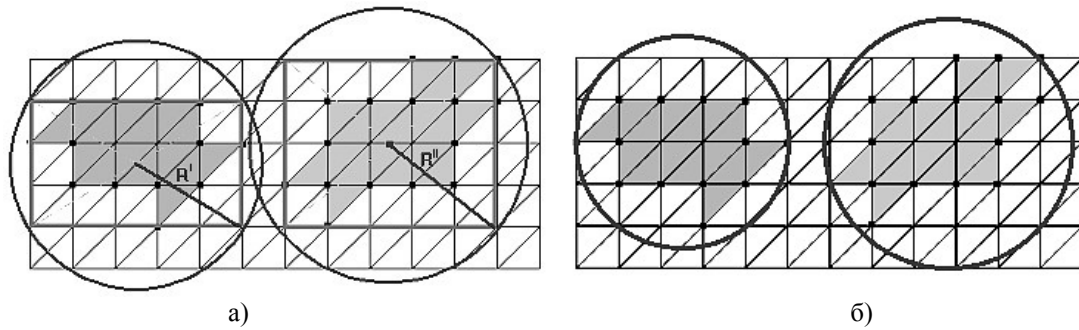


Рис. 8. Габаритные окружности общего вида (а) и редуцированные (б)

принадлежащие границе  $i$ -го препятствия,  $N_i$  - их количество.

**Шаг 1.** Начальная текущая окружность. Вариант 1: возьмем любые две точки  $P_r$ , пусть это точки с  $r = 1, 2$ . Построим на точках  $P_1, P_2$  окружность как на диаметре. Вариант 2: либо иначе - возьмем любые три точки  $P_r$ , не лежащие на одной прямой. Пусть это точки с  $r = 1, 2, 3$ . Построим на них окружность, такая окружность единственна.

Объявим вычисленную окружность текущей. Вычислим координаты ее центра и радиус  $r_c$ .

**Шаг 2.** В цикле для всех точек с  $r \geq 3$  (или  $r > 3$  для Варианта 2 Шага 1) до  $N_i$  последовательно выполним процедуру контроля: проверим, находится ли эта точка внутри, или вне построенной окружности. Эта проверка выполняется сравнением с радиусом текущей окружности  $r_c$  расстояния от центра окружности до очередной точки. Если очередная точка  $P_r$  находится внутри текущей окружности, то продолжение цикла, если вне - переход к Шагу 3, если точки исчерпаны - окончание алгоритма, переход к Шагу 4.

**Шаг 3.** Пересчитаем текущую окружность. Для этого проведем отрезок от  $P_r$  через центр текущей окружности до пересечения с ее дальней от  $P_r$  границей (за центром). Построим новую текущую окружность на этом отрезке, как на диаметре. Вычислим координаты ее центра и радиус  $r_c$ . Переход к Шагу 2.

**Шаг 4.** Окончание алгоритма. Текущая окружность есть искомая редуцированная габаритная окружность. Она по построению обязательно пройдет через некоторую вершину препятствия (здесь отличие от Алгоритма А2). Все точки облака вершин препятствия лежат внутри нее или на ней по построению.

**Алгоритм А4. Габарит - эллипс.** Опишем алгоритм построения (некоторого) габаритного

эллипса для множества вершин препятствия. Этот эллипс также есть гладкая фигура, охватывающая облако граничных вершин препятствия. Строить эллипс будем по трем точкам следующим образом. Пусть точки, на которых будем строить эллипс, есть  $A_1, A_2, A_3$ . Выберем среди них пару точек, расстояние между которыми максимально. Пусть это  $A_1, A_2$ , примем отрезок  $A_1A_2$  за большую ось эллипса  $2a$ . Примем за центр эллипса и полюс полярной системы координат  $O$  середину отрезка  $A_1A_2$ . В этой полярной системе полярный радиус точки  $A_3$  есть вектор между  $O$  и  $A_3$ , длина которого пусть  $\rho$ . Полярный угол  $\varphi$  определяется как угол между известными векторами  $OA_2$  (или  $A_1A_2$ ) и  $OA_3$ . В выбранной полярной системе координат уравнение искомого эллипса имеет вид:

$$\rho = \frac{b}{\sqrt{1-e^2 \cos^2 \varphi}} = \frac{a \sqrt{1-e^2}}{\sqrt{1-e^2 \cos^2 \varphi}}, \quad (3)$$

где  $a, b$  - большая и малая полуоси эллипса;  $e$  - эксцентриситет;  $\rho$  и  $\varphi$  - полярные координаты. Неизвестной величиной является только эксцентриситет  $e$ . Решая второе уравнение (3) относительно  $e$ , получим:

$$e = \sqrt{\frac{\frac{a^2}{\rho^2} - 1}{\frac{a^2}{\rho^2} - \cos^2 \varphi}}. \quad (4)$$

Соотношение (4) завершает построение искомого эллипса. Заметим, что в нем подкоренное выражение всегда положительно ( $a \geq \rho$ ), и  $e \geq 0$ . При  $a = \rho$  и  $\cos \varphi \neq 1$  получим  $e = 0$ , и эллипс переходит в окружность.

**Шаг 1.** Найдем габаритные прямоугольники согласно уравнению (1).



*Шаг 2.* Строить эллипс как в Алгоритме А2, используя диагональ прямоугольника как большую ось эллипса и используя какую-либо третью вершину прямоугольника нецелесообразно, так как такой эллипс совпадет с окружностью по А2. Поэтому увеличим большую ось симметрично относительно центра прямоугольника с коэффициентом  $\gamma$ . Если диагональ прямоугольника равна  $2r$ , выберем большую ось  $2a$  следующим образом:

$$2a = 2r \cdot \gamma, \quad a = r \cdot \gamma, \quad \gamma \geq 1$$

(при  $\gamma = 1$  эллипс перейдет в окружность, но так будем считать лишь для общности).

*Шаг 3.* Выберем три точки для построения эллипса - конечные точки отрезка  $2a$ , проходящего по диагонали прямоугольника (1), и третью вершину прямоугольника, не лежащую на этой диагонали. Построим эллипс на найденных трех точках. Вычислим его координаты центра  $x_0, y_0$ , большую полуось  $a$ , эксцентриситет  $e$ .

Этот эллипс по построению по симметрии обязательно пройдет через четвертую вершину габаритного прямоугольника. Все точки облака вершин препятствия лежат внутри этого прямоугольника или на нем, поэтому они лежат внутри построенного эллипса. Заметим, при построении большая ось эллипса будет направлена вдоль диагонали (одной из двух) габаритного прямоугольника. Соответственно будет повернут эллипс. Следовательно, к введенным выше четырем параметрам эллипса  $x_0, y_0, a, e$  необходимо добавить еще угол  $\vartheta$  между большой осью эллипса и осью абсцисс системы координат карты, в которой строится эллипс. Всего эллипс будет задаваться пятью координатами.

**Алгоритм А5. Габарит - редуцированный эллипс.** Опишем переборный алгоритм построения редуцированного габаритного эллипса. Он линейно зависит от числа вершин препятствия и строит габаритный эллипс итерационным процессом. Этот алгоритм имеет большую вычислительную сложность, чем предыдущий, но позволяет построить эллипс меньшего размера, нежели А4. Эллипс будем строить как сжатую окружность, сжатие при этом выполняется вдоль хорд, ортогональных некоторому диаметру окружности, и проходящих через точки препятствия. Выполним следующие шаги.

*Шаг 1.* Построим габаритную окружность по Алгоритму А2.

*Шаг 2.* Определим коэффициент ее сжатия  $k$ . Для этого повернем исходную систему координат и выполним ее перенос так, чтобы новая ось абсцисс проходила по той диагонали габаритного прямоугольника, на которой построена окружность на предыдущем шаге, пусть это диаметр  $2r$ . В этой системе от всех вершин препятствия проведем перпендикуляры к диаметру  $2r$ , пусть их длины  $d_i, 1 \leq i \leq N_i$  и продлим их до пересечения с окружностью в тех же полуплоскостях, где лежат проверяемые точки, пусть длины полученных полухорд  $h_i, 1 \leq i \leq N_i$ . Коэффициент сжатия

$$k = \min_i \frac{d_i}{h_i}, \quad 1 \leq i \leq N_i, \quad k \neq 0, \quad k \leq 1.$$

*Шаг 3.* Построение эллипса теперь выполняется по следующим формулам:

$$a = r, \quad b = kr, \quad c = \sqrt{a^2 - b^2} = r\sqrt{1 - k^2},$$

$$e = \frac{c}{a} = \sqrt{1 - k^2}, \quad 0 \leq e < 1.$$

Отметим, что по построению большая ось этого эллипса совпадает с диагональю габаритного прямоугольника и эллипс обязательно пройдет хотя бы через одну вершину препятствия, в этом отличие от Алгоритма А4.

**Алгоритм А6. Габарит - минимизированный эллипс.** Рис. 9 показывает, что при необходимости возможно дальнейшее уменьшение построенного габаритного эллипса. Построение минимизированного эллипса как габарита препятствия таково. Заметим сразу, что у такой процедуры вычислительная трудоемкость существенно выше предыдущих.

Параметризуем эллипс следующим образом. Пусть  $x, y$  - центр эллипса, его большая ось составляет угол  $\varphi$  с осью абсцисс, полуоси равны  $a$  и  $b$ , отношение полуосей равно  $k, k = b/a$ , так что  $b = ka$ . Параметр  $k$  связан с эксцентриситетом эллипса  $e$  соотношением  $k^2 + e^2 = 1$ . Тогда для любых заданных  $x, y, \varphi, k$  нетрудно вычислить такое  $a$ , при котором эллипс с параметрами  $x, y, \varphi, a, b = ka$  будет охватывать все точки препятствия. Для этого требуется однократно пройти по всем точкам препятствия  $(x_i, y_i)$ , и для каждой сделать следующее:

- найти радиус-вектор точки  $(x_i - x, y_i - y)$  относительно центра эллипса;
- повернуть этот вектор на угол  $-\varphi$ , чтобы перейти в систему координат, оси которой

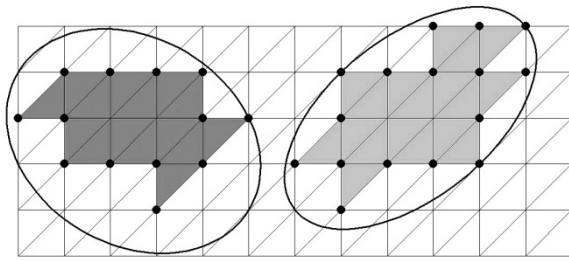


Рис. 9. Габариты – редуцированные эллипсы

параллельны осям эллипса; обозначим повернутый вектор  $(u_i, v_i)$ ;

- чтобы точка лежала внутри эллипса, должно выполняться условие  $u_i^2/a^2 + v_i^2/b^2 \leq 1$ . Его можно переписать в виде  $u_i^2/a^2 + v_i^2/(ka)^2 \leq 1$  или  $u_i^2 + v_i^2/k^2 \leq a^2$ , таким образом, вычислив для каждой точки  $a_i = u_i^2 + v_i^2/k^2$  и вычислив  $a = \max\{a_i\}$ , получим искомый эллипс, охватывающий точки препятствия.

Таким образом, имеется процедура, которая строит охватывающий эллипс для любых заданных значений  $x, y, \varphi, k$ . После этого его можно минимизировать, варьируя  $x, y, \varphi, k$ . Для построения такого минимального эллипса можно использовать любой алгоритм численной минимизации, например, метод Ньютона. В качестве переменных используются  $x, y, \varphi, k$ , в качестве минимизируемой функции можно выбрать площадь эллипса  $\pi ab$ , а в качестве начального приближения, например, эллипс, построенный в алгоритме А5.

Однако отметим еще раз, что вычислительная сложность этой процедуры довольно высока. Метод Ньютона требует на каждом шаге итерации многократного вычисления значения функции (для оценки частных производных), а каждый такой расчет требует прохождения по всему массиву точек для вычисления размера эллипса. Поскольку бортовые системы роботов нередко ограничены в вычислительных ресурсах, и при этом планирование движения должно осуществляться в реальном времени, во многих случаях применение этого алгоритма является излишне сложным, и близких к минимальному и при этом намного проще вычисляемых габаритов А1-А5 вполне достаточно. Но при необходимости (например, при плотном расположении препятствий и необходимости точного нахождения узких проходов) точная минимизация вполне возможна.

**Алгоритм А7. Построение гарантированных габаритов препятствий.** Определенные выше габаритные прямоугольники, окружности или эллипсы заведомо охватывают только те точки (и треугольники), которые строго принадлежат препятствиям. Однако под треугольниками, определенными как теневые (боковые тени и тени за препятствием), могут находиться части препятствий, не видимые дальномером из точки стояния при заданных значениях углов сканирования. Чтобы учесть такую возможность, т.е. гарантированно на 2D-карте отделить области с препятствиями от подстилающей поверхности, можно построить так называемые гарантированные габариты - габариты, охватывающие и препятствия, и их тени. Эти построения могут выполняться в точности как описанные выше, только в рассмотрение необходимо вводить и треугольники препятствий, и треугольники, окрашенные как их тени.

Наконец, завершая описание Этапа 8 общего алгоритма, укажем два критерия возможного сравнения построенных габаритов препятствий. Как уже отмечалось, важным критерием является быстродействие алгоритма вычисления габарита. Здесь наиболее эффективен простой алгоритм построения габаритов-прямоугольников. Другой важный критерий – площадь построенного габарита. Оказывается, что здесь в разных ситуациях выгодны разные алгоритмы. Например, для узких вытянутых препятствий выгоднее оказываются прямоугольные габариты, для иных препятствий – охватывающие окружности или эллипсы.

### 3. Примеры работы алгоритма

На Рис. 10 показано, как алгоритм определяет лежащий перед роботом параллелепипед («кубик») на плоскости. Дальномер расположен в точке  $D$ , зоны  $A, B, C$  – разные зоны препятствия, зона  $A$  – само препятствие, зоны  $B, C$  – тени вокруг препятствия.

На Рис. 11 дана аналогичная картина для находящейся перед роботом полусферы, буквенные обозначения такие же, как на Рис. 10.

На Рис. 12 показана карта (сверху), на которой нанесены 4 препятствия. На карте представлены результаты обработки измерений одного скана в проекции на плоскость  $D'x'y'$  (ось  $x'$  направлена горизонтально, а ось  $y'$  – вертикально). Точка  $D'$  соответствует проекции

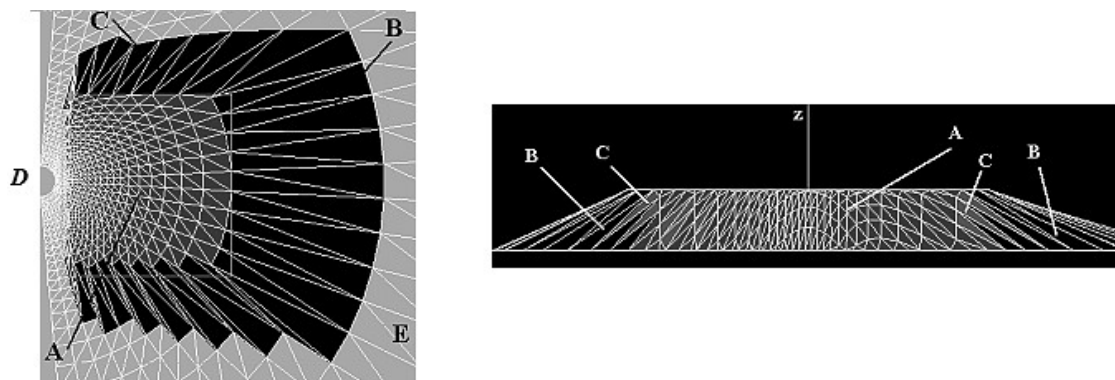


Рис. 10. Вид «кубика» на плоскости

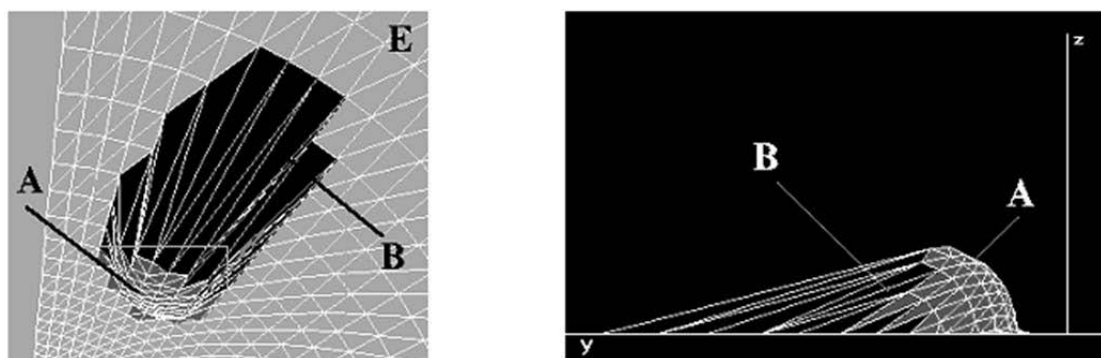


Рис. 11. Вид полусферы на плоскости

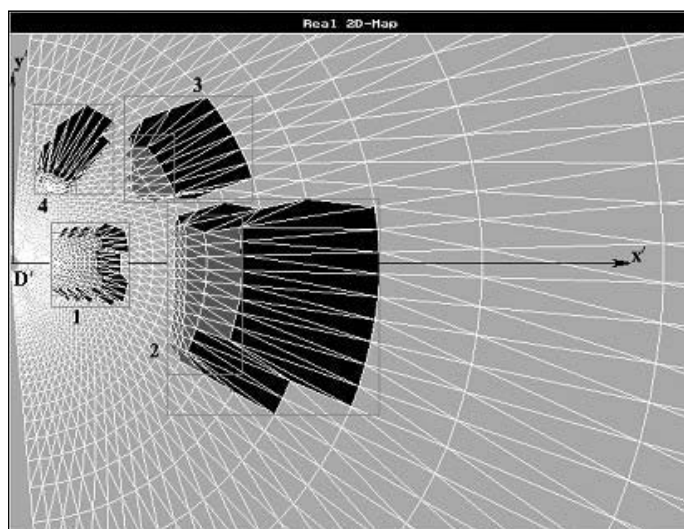


Рис. 12. Карта препятствий

точки подвеса дальномера и является началом полярной сетки, определяющей сканирование. На карте также показаны габариты-прямоугольники найденных препятствий.

При этом номерами 1 и 3 обозначены проекции измерений двух параллелепипедов (и их теней), определенных системой как препятствия. Объект с номером 2 представляет собой

проекцию измерений призмы с основаниями, параллельными плоскости  $D'x'y'$ , и с наклоненной стороной, параллельной оси  $y'$ . Заметим, что угол наклона этой стороны больше предельного угла наклона поверхности, являющейся проходимой для робота (так были заданы соответствующие значения параметра  $c\theta$  и координаты вершин этой призмы), поэтому

данная сторона была определена как препятствие. Объект с номером 4 - это проекция измерений полусферы, центр которой лежит на плоскости сцены.

## Заключение

В ходе исследований различных примеров проведены эксперименты по использованию описанного алгоритма для чистых и зашумленных данных, приближенных к данным реального устройства. Эти эксперименты показали устойчивую работу алгоритма в пределах ошибок порядка 1-1,5% от величины исходной измеренной дальности. В указанном диапазоне алгоритм устойчиво выделял препятствия, причем их размеры и границы мало отличались от случая идеальных точных данных. При больших значениях ошибок алгоритм также выделял реальные препятствия, но при этом появлялись дополнительные «фантомные» препятствия - зоны, которые обозначались алгоритмом как препятствия, хотя фактически являлись проходимыми участками.

Проведенные эксперименты позволяют сделать вывод о возможности применения данного алгоритма в 1,5%-ном диапазоне точности измерений. Кроме того, алгоритм показал достаточную эффективность в вычислительном плане, позволяющую реализовать его в бортовой системе. Эти эксперименты будут продолжены.

Автор благодарит коллег из лаборатории "Интеллект и мехатроника" ИПМ им. М.В. Келдыша РАН и РГГУ за помощь в работе и обсуждения, а также В.В. Павловского за большую помощь в проведении исследований и разработку программы картирования препятствий. На начальном этапе работы существенный вклад внесла Ю.Н. ИONOва, автор выражает ей искреннюю благодарность.

**Павловский Владимир Евгеньевич.** Профессор, главный научный сотрудник Института прикладной математики им. М.В. Келдыша РАН. Окончил механико-математический факультет МГУ им. М.В. Ломоносова в 1972 году. Доктор физико-математических наук. Автор 320 печатных работ и двух монографий. Область научных интересов: робототехника, методы управления роботами. E-mail: vlpavl@mail.ru

## Литература

1. Электронный ресурс: <http://www.smprobotics.ru/ustroystvo-lidarov/>
2. Электронный ресурс: <http://robotgeeks.ru/blogs/articles/sravnenie-harakteristik-lazernyh-skanerov-sick-lms-200-i-hokuyo-urg-04-ug01>
3. Лемтюжников Д.С. Элементарный курс оптики и дальномеров. М.: Воениздат. 1938, с. 136.
4. Лысенко О.Н. Использование лазерных сканеров SICK AG для навигации мобильных роботов, // Компоненты и технологии. №1'. 2008, с.56-59.
5. Электронный ресурс: <https://www.youtube.com/watch?v=NDKSG2v8His>
6. Электронный ресурс: [http://www.rusnauka.com/20\\_PNR\\_2011/Tecnic/11\\_90438.doc.htm](http://www.rusnauka.com/20_PNR_2011/Tecnic/11_90438.doc.htm)
7. Dickmanns E D., Zapp A. Autonomous High Speed Road Vehicle Guidance By Computer Vision. // Proc. of International Federation of Automatic Control. World Congress (10th), 1988, pp. 221-226. (<https://trid.trb.org/view.aspx?id=494179>).
8. Электронный ресурс: <https://rg.ru/2015/06/04/kamaz-site-anons.html>
9. Электронный ресурс: <http://www.tartanracing.org/blog/>
10. Павловский В.Е., Огольцов В.Н., Огольцов Н.С. Микрокомпьютерная система управления нижнего уровня для автомобиля с механической трансмиссией. // Мехатроника. Автоматизация. Управление. №6. 2014, 29-36.
11. Павловский В.Е., Огольцов В.Н., Спиридонова И.А., Павловский Е.В. Задачи управления беспилотным автомобилем, экспериментальная реализация в проекте "АвтоНИВА". // Робототехника и техническая кибернетика. СПб. ГНЦ ЦНИИ РТК. №4 (9). 2015, с.41-46.
12. Павловский В.Е., Огольцов В.Н., Спиридонова И.А. Задачи управления беспилотным автомобилем в проекте "АвтоНИВА". // Тр. Международной научно-технической конференции "Экстремальная робототехника". / Тр. II Всероссийского семинара "Беспилотные транспортные средства с элементами искусственного интеллекта (БТС ИИ)" (2015). Изд-во "Политехника-Сервис". СПб. 2015, с.107-114.
13. Al-Sultan K.S., Aliyu M.D.S. A new potential field-based algorithm for path planning. // Journal of Intelligent and Robotic Systems. November 1996, Volume 17, Issue 3, pp 265-282. DOI:10.1007/BF00339664.
14. Hart P.E., Nilsson N.J., Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. // IEEE Transactions on Systems Science and Cybernetics SSC4. 4 (2): 100-107. DOI:10.1109/TSSC.1968.300136.

## **Heuristic algorithm of isolated obstacles detection by mobile robot according to ranging data**

V.E. Pavlovsky

**Abstract.** The algorithm solving a problem of detection of the single isolated obstacles by mobile robot by means of a range finder is described. The main unit of algorithm is constructed as system of productional rules which are introduced by the logical ratios allowing to find existence or lack of an obstacle across the field of normals to a surface. The found obstacles are brought on the two-dimensional map, methods of mapping of obstacles are discussed.

The paper is prepared as expanded version of the report submitted on the Conference on Artificial Intelligence CAI-2016, Smolensk, 2016.

**Keywords:** mobile robot, range finder, productional system

**Vladimir E. Pavlovsky.** Keldysh Institute of Applied Mathematics of RAS, Moscow, Russia. E-mail: vlpavl@mail.ru