

This article describes methods of solving problems of automatic navigation of robotized mobile devices inside of a building. A robotized mobile device has tracks and can move back and forth and turn right-hand and left-hand. Micro controller based on Arduino is responsible for receiving and processing of information taken from sensors. Analysis of the data occurs directly onboard of a robotized mobile device using a single-board computer Raspberry Pi or some kind of it. The article shows a method and an algorithm of a robotized mobile device to follow the path and then make a map of a particular building. While moving a robotized mobile device records in memory the data received from sensors. Then it processes the data and forms a map of the building. The relevance of this work is that the most common navigation systems using GPS (Global Positioning System) / GLONASS (GLobal NAVigation Satellite System) don't work in those conditions when a robotized mobile device is located in buildings, such as a room or a building hallway. A satellite signal is lost while going through cemental, metal or wooden constructions. The algorithms presented in the article are based on the properties of the standard deviation values of the distance to objects surrounding a robotized mobile device. The data were obtained through a set of range finders, which was installed onboard of a robotized mobile device, without using a complex and expensive scanning device. Developed algorithms have an advantage, they can work in a building without reference to GPS or GLONASS. The main features of the algorithm are simplicity, low price and accessibility of the used hardware.

**Keywords:** robotized mobile devices, map of a building, dispersion, the standard deviation, a set of rangefinders, pathfinder

For citation:

**Smirnov A. V., Bezzubtcev A. Yu.** Automatic Moving of a Robotic Mobile Device Making a Map of a Building, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2016, vol. 17, no. 6, pp. 395–400.

DOI: 10.17587/mau.17.395-400

### References

1. **SLAM** — *principy i slylkinia open source* (SLAM — principles and references to open source), available at: <http://my-it-notes.com/20.13/01/slam-basis-and-links-at-open-source/> (accessed 17.07.2015).
2. **SLAM**. Robot-develop, available at: <http://robot-develop.org/archives/3954> (accessed 17.07.2015).
3. **Delphi skanirujushij lazernyj dal'nomer** [Scanning range finder]. Robot-develop, available at: <http://robot-develop.org/archives/3700> (accessed 17.07.2015).
4. **Lidar** [Light Detection and Ranging], available at: <http://systemsauto.ru/active/lidar.html> (accessed 17.07.2015).
5. **Minin A. A.** Navigacija i upravlenie mobil'nym robotom, osnashennym lazernym dal'nomerom: Avtoref. dis. kand. tehn. nauk, Moskva, 2008, 18 p.
6. **Ul'trazvukovoj datchik izmerenija rasstojanija HC-SR04** (Ultrasonic ranging module HC-SR04), available at: <http://robocraft.ru/blog/electronics/772.html> (accessed 17.07.2015).
7. **IK-dal'nomery SHARP** (IR range finders), available at: [http://roboforum.ru/wiki/ИК-дальномеры\\_SHARP](http://roboforum.ru/wiki/ИК-дальномеры_SHARP) (accessed 17.07.2015).
8. **Datchik izmerenija rasstojanija SHARP-GP2Y0A710K0F — raschj ot rasstojanija** (The sensor measurement range SHARP-GP2Y0A710K0F — calculation of distance), available at: <http://robocraft.ru/blog/arduino/770.html> (accessed 17.07.2015).
9. **Soga Laser Ranging Module Preview**, available at: <http://www.electronics.com/7180.html> (accessed 17.07.2015).
10. **Computer vision**, available at: <http://robodem.ru/machinevision> (accessed 17.07.2015).
11. **Arduino Uno**, available at: <http://arduino.ru/Hardware/ArduinoBoardUno> (accessed 17.07.2015).
12. **Raspberry Pi 2 modelB**, available at: <http://www.raspberrypi.org/products/raspberrypi-2-model-b/> (accessed 17.07.2015).
13. **Orlov A. I.** *Dispersija sluchajnoj velichiny, Matematika sluchajno: Veroyatnost' i statistika — osnovnyefakty* (The variance of the random variable, Mathematics cases: Probability and statistics — key facts), Moscow, Progress, 2003, 2004 (in Russian).
14. **Borovikov V.** *Statistica. Iskusstvo analizadannyh na komp'yutere: Dlja professionalov* (The art of data analysis on the computer: For professionals), Peterburg, Progress, 2003, 688 p. (in Russian).
15. **Rabota s magnitometrom HMC5883L** (Working with the magnetometer HMC5883L), available at: [http://avrproiect.ru/publ/kak-podkljuchit-rabota\\_s-magnitometrom-hmc5883l-2-1-0-150](http://avrproiect.ru/publ/kak-podkljuchit-rabota_s-magnitometrom-hmc5883l-2-1-0-150) (accessed 17.07.2015).
16. **Akselerometr i giroskop MPU6050** (Accelerometer and gyroscope MPU6050), available at: <http://cxem.net/mc/mc324.php> (accessed 17.07.2015).
17. **Odometer**, available at: <http://autodrop.ru/stati/517-chto-takoe-odometr.html> (accessed 17.07.2015).
18. **Encoder**, available at: <http://robocraft.ru/blog/technology/734.html> (accessed 17.07.2015).

УДК [004.896: 004.932]: [681.586: 519.712]

DOI: 10.17587/mau.17.400-406

**Нгуен Туан Зунг**, науч. сотр., [dunghvkt@yahoo.com](mailto:dunghvkt@yahoo.com),

Вьетнамская академия наук и технологий,

**И. А. Щербатов**, канд. техн. наук, доц., [sherbatov2004@mail.ru](mailto:sherbatov2004@mail.ru),

**О. М. Проталинский**, д-р техн. наук, проф., [protalinskiy@gmail.com](mailto:protalinskiy@gmail.com),

Астраханский государственный технический университет

## Система управления мобильным роботом на основе модифицированного алгоритма 3D-pointcloud

Предложена система управления мобильным роботом в зданиях и сооружениях, в основу которой положена методика построения траекторий движения с использованием сенсора глубины датчика Kinect. Синтезирован модифицированный алгоритм 3D-pointcloud, обеспечивающий решение задачи перемещения робота без столкновений с препятствиями. Показаны результаты экспериментов, подтверждающие эффективность изложенного в работе подхода.

**Ключевые слова:** датчик глубины, робот, SURF, 3D-pointcloud, распознавание образов, библиотека FLANN, система управления, траектория робота, предотвращение столкновений робота, навигация

## Введение

Одной из основных задач управления мобильными роботами является необходимость объезда препятствий. При управлении движением робота используются системы навигации, базирующиеся на различных датчиках (например, стереокамеры [1, 2], электронный компас, инклинометр, сонар [3] и т. д.). Функциональность применяемых подходов зависит от эффективности и быстродействия алгоритмов распознавания и отслеживания препятствий. Для этих целей обычно применяются два класса методов: распознавание объектов на основе RGB-изображений (метод 1) [4–6] и пространственно-временной (метод 2) [7, 8]. Первый класс методов во многих случаях приводит к снижению точности распознавания и увеличению вычислительных мощностей из-за различий в изменении условий освещения и углов зрения средств фиксации изображения. Для устранения указанных недостатков применяется второй класс методов, обеспечивающих обнаружение препятствий на основе характерных областей изображения.

При навигации мобильных роботов планирование пути — это определение оптимальной траектории между начальной и целевой точками без столкновения с препятствиями [9, 10]. Как правило, траектория разделяется на линейные сегменты. В фазе навигации робот перемещается по этим сегментам [3]. Для вычисления безопасных кратчайших траекторий предложены несколько методов, использующихся при локальном или глобальном планировании движения [1]. Планирование глобального пути учитывает всю информацию об окружающей среде, и для этих целей применяется ряд методов: планирование Вороного [11, 12], сотовая декомпозиция (cell decomposition) [13] и т. д. Реализация этих подходов требует значительного времени на этапе предварительного расчета и использования вспомогательного аппаратного оборудования для увеличения быстродействия [12]. Основной недостаток — необходимость создания базы образов препятствий окружающего пространства для построения его топологии и связности объектов.

Алгоритмы локального планирования разработаны для предотвращения столкновения с препятствиями в непосредственной близости от робота. Для этого используется только информация о ближайших препятствиях. Наиболее известными являются методы на основе потенциальных полей [14], в которых пространство разделено регулярными сетками, на основе которых обеспечивается поиск оптимального пути без столкновения с препятствиями. Эти методы являются достаточно быстродействующими, но не позволяют предотвратить попадание в бесконечный цикл (локальные минимумы потенциальной функции). Для исключения проблемы локальных минимумов и получения оптимального пути используется подход, описанный

в работе [2] и обеспечивающий нахождение "грубой" траектории с последующим уточнением методами оптимизации.

При управлении роботом существует ряд трудностей в обнаружении препятствий и применении алгоритма планирования движения. Для их устранения необходимо разработать методику, использующую информацию о глубинах изображений, например от датчика Kinect, и алгоритм навигации робота для перемещения без столкновений. Для этого требуется решить три взаимосвязанные подзадачи: реализовать алгоритм 3D-Pointcloud для обнаружения любых препятствий в ближней зоне робота и алгоритм SURF для распознавания ориентиров; синтезировать методику построения карты препятствий; синтезировать алгоритм управления роботом.

## Общий алгоритм работы системы

Датчик Kinect позволяет получать карту глубин изображения и имеет ряд преимуществ перед другими датчиками, работающими при низком уровне освещенности, позволяя получать калиброванную масштабную оценку, инвариантную по отношению к цвету и текстуре. Получаемые изображения глубины имеют разрешение  $640 \times 480$  пикселей при 30 кадрах в секунду. Схема работы системы управления роботом показана на рис. 1. Методика включает три основных этапа: обнаружение препятствий, построение карты препятствий и построение траекторий.

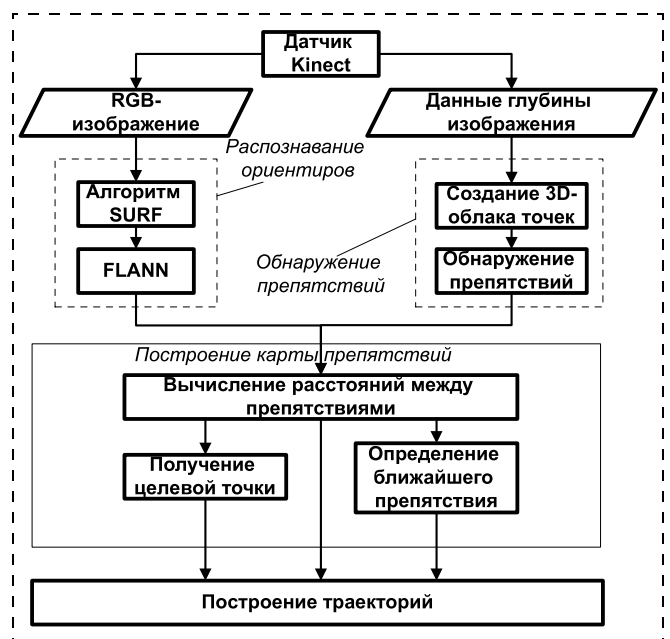


Рис. 1. Общая схема методики управления роботом

## Распознавание ориентиров и обнаружение препятствий

Распознавание ориентиров (объектов) в потоке видеоданных является сложной задачей, для решения которой обычно применяются алгоритмы, использующие статистические выборки на основе локальных особенностей (например, HOG — Histograms of Oriented Gradients, EOH — Edge orientation histograms [15]), а также алгоритмы на основе извлечения особых точек изображения (например, SIFT — Scale Invariant Feature Transform [6], SURF — Speeded Up Robust Features [5]).

Алгоритм SURF является более предпочтительным по сравнению с другими [4]. Эффективность нахождения объектов с использованием алгоритма SURF можно увеличить за счет совместного применения библиотеки FLANN (рис. 2, 3, см. вторую сторону обложки) [16]. Требуемые скорость и точность распознавания также обеспечиваются и при движении самого робота. Эффективность обнаружения препятствий (подвижных и неподвижных объектов) в режиме реального времени с применением 3D-облака точек доказана в работе [17].

Облака точек — набор неорганизованных иррегулярных точек в трехмерном пространстве. Облака точек хранят координаты всех точек в ближней зоне мобильного робота, а также RGB-цвета, значения интенсивности и т. д. 3D-облака точек могут быть получены из карты глубины изображения, полученной от датчика Kinect (рис. 4).

Эффективность данной схемы выше, чем традиционных алгоритмов, которые базируются на изображениях, полученных из видимого спектра, по аналогии с тем, как это воспринимается человеческим глазом [18]. При этом применение алгоритма понижения разрешения и DBSCAN (плотностный алгоритм кластеризации пространственных данных с присутствием шума) [19] существенно увеличивает быстродействие алгоритма распознавания объектов и повышает его точность.

## Построение карты препятствий

Окружающая среда робота очень сложна из-за различий при изменении условий освещения, углов зрения и наличия препятствий (подвижных и неподвижных). Кроме того, состояние окружающей среды может изменяться достаточно быстро, что приводит к снижению точности управления. Поэтому построение карты препятствий является ключевым этапом. Для перемещения без столкновений робот должен обладать информацией о препятствиях (например, о текущем местоположении, размерах



Рис. 4. Общая схема обнаружения препятствий

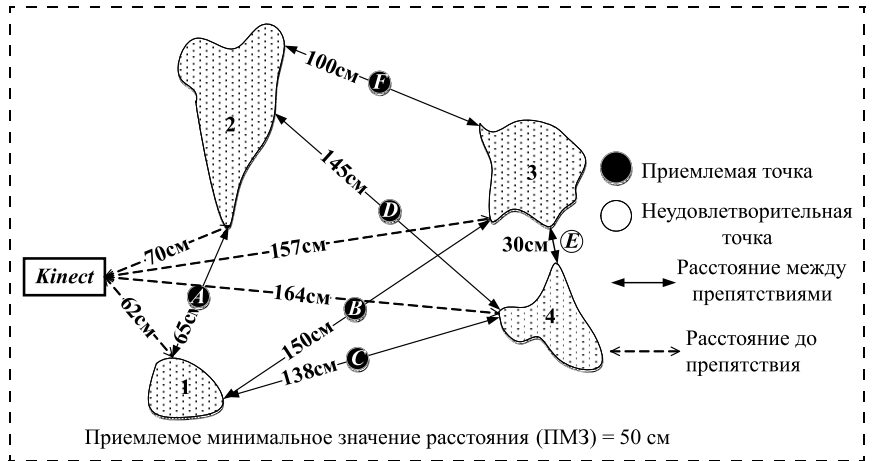


Рис. 5. Пример построения карты препятствий

препятствий и т. д.). Построение карты препятствий показано на рис. 5.

- Алгоритм обеспечивает решение трех подзадач:
- получение информации о препятствиях;
  - вычисление расстояний от датчика Kinect до препятствий и наименьших расстояний между препятствиями;
  - получение требуемых точек: получение границ препятствия;
  - сортировка расстояний от датчика Kinect до каждого препятствия в порядке от ближайшего к наиболее удаленному, а также формирование спи-

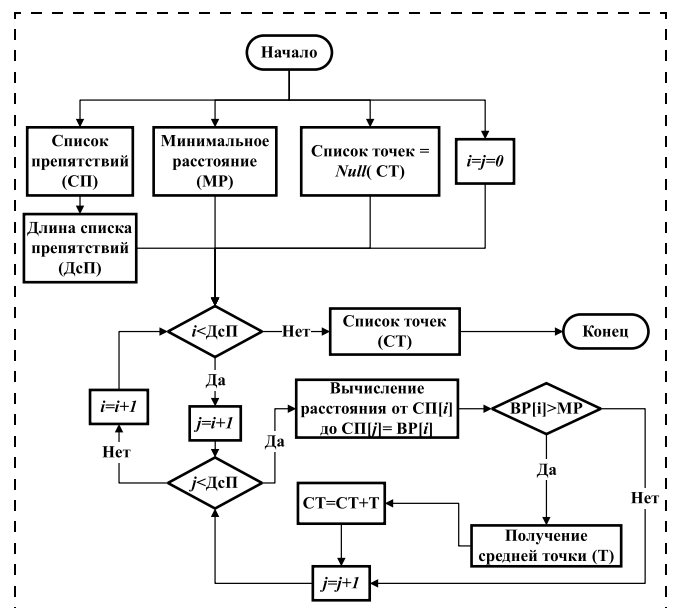


Рис. 6. Блок-схема алгоритма построения карты препятствий

ска препятствий (множества СП); вычисление расстояний от любого препятствия до остальных препятствий (получение множества СР); вычисление средней точки линии между препятствиями на основе множества СР. Если длина этой линии больше, чем приемлемое минимальное значение (ПМЗ), то средняя точка является допустимой (точки *A, B, C, D, F* на рис. 5), в противном случае — нет (точка *E*). Все допустимые точки сохраняются в множестве точек СТ. Реализующий данный подход алгоритм представлен на рис. 6.

### Построение траекторий движения робота

Препятствия представляются многоугольниками на основе изображений, получаемых от датчика Kinect, и имеют реальные геометрические размеры, что наряду со свободными областями пространства позволяет вычислить траекторию перемещения робота из начальной точки в целевую.

Вычисление расстояния между препятствиями осуществляется по паре антипод между полигонами, представляющими препятствия, для случаев "вершина—вершина"; "вершина—граница"; "граница—граница".

Таким образом, выбор точки, в которую следует переместиться, обусловлен расстоянием до препятствия и наличием "свободного" прохода, т. е. расстояния между препятствиями, превышающего геометрические размеры фронтального сечения робота.

Построение траектории — поиск минимального пути от начальной точки до целевой, характеризующегося стоимостью пути. Стоимость пути — расстояние, которое проходит робот из текущего положения в целевое (сумма длин отрезков траектории между ними).

В общем случае стоимость пути является произвольной функцией дискретного пути —  $F(P)$ . Тогда стоимость пути — сумма стоимостей участков траектории, пройденных до точки  $p_i$ ,  $G(p_i)$  и стоимости перемещения от текущей точки до следующей  $H(p_i, p_{i+1})$ :

$$F(P) = G(p_i) + H(p_i, p_{i+1}).$$

Обе функции  $G$  и  $H$  могут быть произвольными. Для типичных ситуаций функция  $G$  будет представлять собой стоимость прохождения через заданные точки к текущей точке  $p_i$ , а  $H$  — евклидово расстояние между точками  $p_i$  и  $p_{i+1}$ .

Для нахождения минимальной стоимости пути обычно применяется алгоритм "А\*" (рис. 7), используемый для поиска пути при обходе графа или в процессе составления эффективного плана прохождения пути между точками и т. д. В общем случае алгоритм находит путь от начального узла к узлу назначения. Эвристическая функция используется для классификации каждого узла в соответствии с оценкой оптимального маршрута через этот узел, т. е. просматриваются узлы в порядке эвристической оценки.

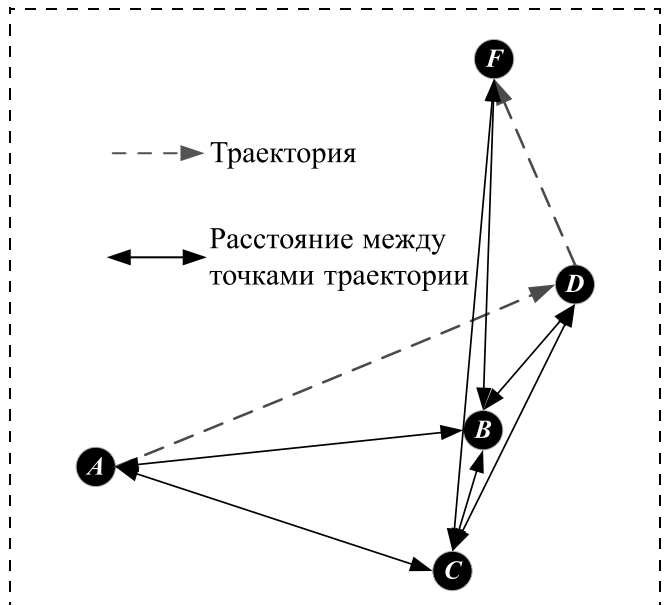


Рис. 7. Применение алгоритма А\* для построения траекторий

Алгоритм просматривает все траектории от начальной вершины до конечной в целях поиска траектории, минимальной по стоимости. На каждом этапе обрабатывается множество траекторий из начального узла до всех нераскрытых вершин, размещающихся в очереди с приоритетом, определяемым по значению:

$$f(n) = g(n) + h(n);$$

$$g(n) = \sum_{i=1}^n d(i); d(i) = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2};$$

$$h(n) = \sqrt{(x_n - x_0)^2 + (y_n - y_0)^2},$$

где  $g(n)$  — стоимость пути от начальной точки к точке  $n$ ;  $h(n)$  — расстояние до цели (или до начальной точки) по прямой линии;  $x_0, y_0$  — координаты начальной точки в плоскости перемещения;  $d(i)$  — евклидово расстояние между точками с координатами  $x_i, y_i$  и  $x_{i-1}, y_{i-1}$  (текущей и предшествующей  $i$ -му моменту времени);  $x_n, y_n$  — координаты целевой точки.

Представим синтезированный алгоритм в следующем виде:

```

создать открытый перечень узлов, содержащий
только начальный узел;
создать закрытый пустой перечень узлов;
while (не достигнута цель)
{ рассмотреть возможность лучшего узла в открытом
списке (узел с наименьшим значением  $f$ )
  If (текущий узел является целью)
  { конец
  }
  Else
  { переместить текущий узел в закрытый список и
рассмотреть всех его соседей

```

**For** (каждый сосед)

**If** (этот сосед находится в закрытом списке и текущее значение  $g$  меньше)

{ обновить соседа с новым, более низким значением  $g$ ; изменить родительский узел соседа на текущем узле

}

**Elseif** (этот сосед находится в открытом списке и текущее значение  $g$  меньше)

{ обновить соседа с новым, более низким значением  $g$ ; изменить родительский узел соседа на текущем узле

}

**Else** (сосед не в открытом или закрытом списке)

{ Добавить соседа к открытому списку и установить его со значением  $g$

}

}

}

Алгоритм работает пока значение  $f(\cdot) = \min$  для целевой вершины, т. е. обеспечено решение с наименьшей стоимостью: чем меньше  $h(\cdot)$ , тем выше приоритет. Поэтому для реализации очереди используются сортирующие деревья.

### Предотвращение столкновений с препятствиями

После построения карты препятствий окружающей среды и создания предварительно вычисленной траектории робот начинает движение в момент получения данных от датчика Kinect. Эти данные непрерывно превращаются в облако точек. Если траектория пересекается с новыми препятствиями или расстояние до препятствий меньше, чем ПМЗ, то робот останавливается. После этого текущее облако точек добавляется к карте препятствий и создается новая карта окружающей среды. После получения новой карты препятствий алгоритм вычисляет другой короткий безопасный путь от текущей позиции робота к цели. Однако предыдущий путь сохраняется, и если препятствия, блокирующие перемещения, исчезают, то робот продолжает движение в соответствии с предыдущей траекторией, т. е. алгоритм не зависит от динамических изменений окружающей среды.

### Эксперименты и анализ результатов

Эксперименты проводили на ЭВМ со следующими характеристиками: центральный процессор Intel Pentium Duo T2390 с частотой 1,87 ГГц, объем оперативной памяти 1 Гб, объем видеопамати 256 Мб. Программная реализация обеспечивается применением библиотек FLANN, OpenCV и EMGUCV в среде Visual Studio C# 2010.

Условия окружающей среды для выполнения экспериментов: робот движется внутри зданий, параметры окружающей среды — минимальное рас-

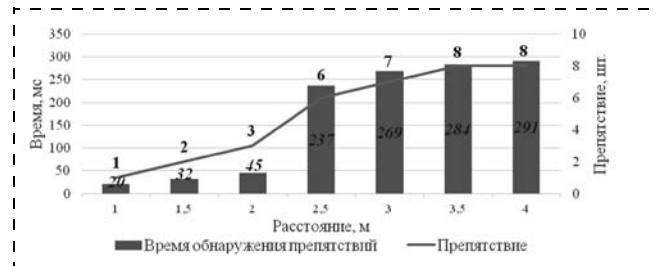


Рис. 9. Результаты применения алгоритма "3D-pointcloud"

### Результаты экспериментов по управлению роботом

Параметры	Значения							Среднее
	1	1,5	2	2,5	3	3,5	4	
Расстояние, м	1	1,5	2	2,5	3	3,5	4	2,5
Препятствие, шт.	1	2	3	6	7	8	8	5
Время обнаружения препятствий (1)	20	32	45	237	269	284	291	168,29
Время построения карты препятствий (2)	8	7	20	183	279	346	356	171,29
Время создания траектории (3)	2	3	8	40	58	62	53	32,286
Общее время (1 + 2 + 3)	30	42	73	460	606	692	700	371,86

стояние от робота 0,4 м, максимальное расстояние от робота 8 м, высота 2 м; в окружающей среде много препятствий (существуют подвижные и неподвижные объекты), их число изменяется; условия освещения и углы зрения изменяются быстро. Предлагаемая методика обнаружения препятствий использует алгоритм 3D-pointcloud и обеспечивает успешное обнаружение препятствий даже при условии отсутствия хорошего освещения (рис. 8, см. вторую сторону обложки).

Из рис. 9 и таблицы следует, что производительность алгоритма является наиболее высокой, если максимальное расстояние до препятствия составляет менее 2,5 м. Время управления движением робота разделяется на три основных составляющих (см. таблицу): время обнаружения препятствий (ВОП); время построения карты препятствий (ВПК); время создания траектории (ВСТ):

$$OV = VOП + ВПК + ВСТ.$$

Данные экспериментов показывают, что для построения траектории движения робота следует применять алгоритм 3D-Pointcloud для обнаружения препятствий и создания карты препятствий, а для построения траектории движения робота можно использовать алгоритм "А\*". Для удобства сравнения полученных в рамках исследования результатов все полученные данные сведены к одной диаграмме, демонстрирующей эффективность предложенной в работе методики и алгоритмов (рис. 10, см. вторую сторону обложки).

## Заключение

Данные экспериментов показывают, что для построения траектории движения следует использовать алгоритм 3D-Pointcloud (для обнаружения препятствий и создания карты препятствий) совместно с алгоритмом "А\*". Структурная схема системы управления роботом, использующая синтезированную методику создания траекторий движения робота на основе данных, получаемых от датчика Kinect, и разработанные модифицированные алгоритмы, показана на рис. 11.

Разработанные алгоритмы могут применяться для любых дальномеров и стереокамер, которые возвращают цветные кадры и изображения глубины, так как описанные алгоритмы используют данные об удаленности объектов от сенсора и устойчивы к видимому спектру света (могут функционировать в полной темноте, не требуя подсветки).

### Список литературы

1. Hwang Y., Ahuja N. Gross motion planning — a survey // ACM Computing Surveys. 1992. 24 (3). P. 219—291.
2. Cunha S., de Matos A., Pereira F. An automatic path planning system for autonomous robotic vehicles Pereira // Proc. of International Conference on Industrial Electronics, Control, and Instrumentation (IECON '93). 1993. Vol. 3, Maui, HI, USA. P. 1442—1447.
3. Hassouna Sabry M., Alaa E. Abdel-Hakim, Aly A. Farag. PDE-based robust robotic navigation // Computer Vision and Image Processing Laboratory (CVIP), Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY 40292, USA. 2007. P. 10—18.
4. Juan L., Gwon O. A Comparison of SIFT, PCA-SIFT and SURF // International Journal of Image Processing (IJIP). 2009. Vol. 3, No. 4. P. 143—152.
5. Bay H., Ess A., Tuytelaars T., Van Gool L. Speeded-Up Robust Features (SURF) // Computer Vision and Image Understanding, June 2008. Vol. 110, Iss. 3. P. 346—359.
6. Ramisa A., Vasudevan S., Aldavert D., Toledo R., Lopez de Mantaras R. Evaluation of the SIFT Object Recognition Method in Mobile Robots. URL: [http://www.iiia.csic.es/~mantaras/sift\\_eval.pdf](http://www.iiia.csic.es/~mantaras/sift_eval.pdf).
7. Носков В. П., Носков А. В. Навигация мобильных роботов по дальнометрическим изображениям // Мехатроника, автоматизация, управление. 2005. № 12. С. 16—21.
8. Mojtahedzadeh R. Robot Obstacle Avoidance using the Kinect. Master of Science Thesis Stockholm, Sweden, 2011. 61 p.
9. Murphy R. Introduction to AI Robotics. MIT press, 2000. 400 p.
10. Bay H., Ess A., Tuytelaars T., Van Gool L. Path planning using lazy prm // Proc. of the IEEE International Conference on Robotics and Automation (ICRA). 1997. P. 333—338.
11. Rao N. S. V., Hareti S., Shi W., Iyengar S. Robot navigation in unknown terrains: introductory survey of non-heuristic algorithms // Oak Ridge National Laboratory, Technical Report ORNL/TM-12410. 1993.

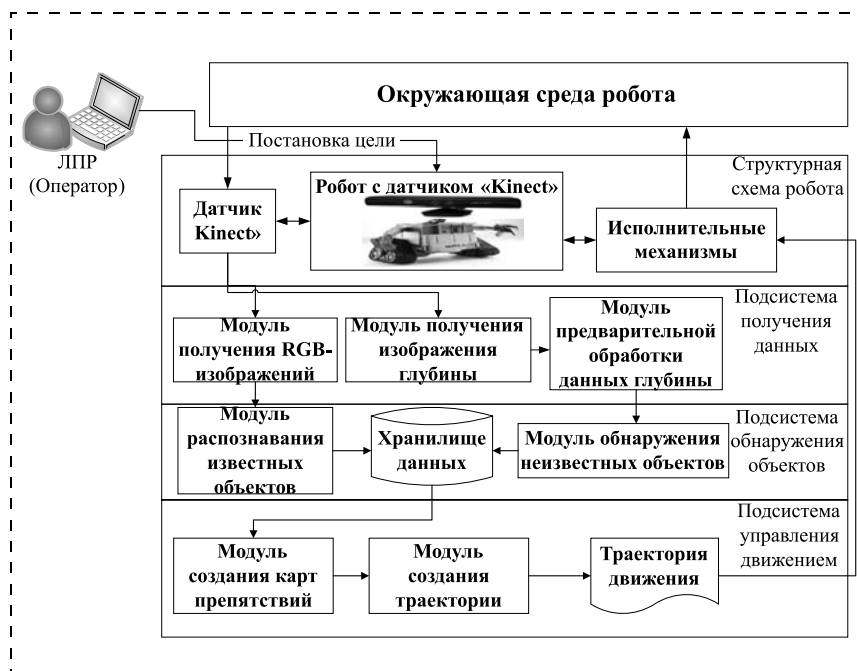


Рис. 11. Структурная схема системы управления роботом

12. Hoff K., Culver T., Keyser J., Lin M., Manocha D. Interactive motion planning using hardware-accelerated computation of generalized Voronoi diagrams // Proc. of the IEEE International Conference on Robotics and Automation. 2000. P. 2931—2937.
13. Zhu D., Latombe J. New heuristic algorithms for efficient hierarchical path planning // IEEE Transaction on Robotics and Automation. 1991. 1 (1). P. 9—20.
14. Hwang Y. K., Ahuja N. Potential field approach to path planning // IEEE Transaction on Robotics and Automation. 1992. 8 (1). P. 23—32.
15. Navneet Dalai, Bill Triggs. Object Detection using Histograms of Oriented Gradients. URL: <http://www.webcitation.org/6DvoEuAvL>.
16. Нгуен Туан Зунг, Щербатов И. А. Распознавание объектов в системе технического зрения мобильного робота: использование библиотеки FLANN и алгоритма SURF // Прикаспийский журнал управление и высокие технологии. — 2014. № 4. С. 65—76.
17. Нгуен Туан Зунг, Щербатов И. А. Совместное распознавание подвижных и неподвижных объектов в системе технического зрения робота // Мехатроника, автоматизация, управление. 2015. Т. 16, № 7. С. 464—470.
18. Shcherbatov I. A., Nguyen Tuan Dung. Speed up algorithms for obstacles detection in the robot vision system using point cloud // Creativity in Intelligent Technologies and Data Science. 2015. P. 673—683. DOI: 10.1007/978-3-319-23766-4\_53.
19. Chakraborty S., Nagwani N. K., Dey L. Performance Comparison of Incremental K-means and Incremental DBSCAN Algorithms // International Journal of Computer Applications (0975 — 8887). August 2011. Vol. 27, No. 11. P. 14—18.
20. Рассел С. Дж., Норвиг П. Искусственный интеллект: современный подход / Пер. с англ. и ред. К. А. Птицына. М.: Вильямс, 2006. 1408 с.
21. Проталинский И. О., Елизаров Д. В., Кирилин С. А. Универсальная мобильная платформа для роботов, обслуживающих социальную и бытовую сферы // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2011. № 2. С. 49—54.

# Mobile Robot Control System Based on a Modified 3D-Pointcloud Algorithm

**Nguyen Tuan Dung**, dunghvkt@yahoo.com, **I. A. Shcherbatov**, sherbatov2004@mail.ru✉, **O. M. Protalinskii**, protalinskiy@gmail.com, Astrakhan State Technical University, Astrakhan, 414056, Russian Federation

Corresponding author: **Shcherbatov Ivan A.**, Ph. D., Associate Professor, Astrakhan State Technical University, Astrakhan, 414056, Russian Federation e-mail: sherbatov2004@mail.ru

Received on January 20, 2016

Accepted on January 29, 2016

The control system of mobile robot (functioning in buildings and structures), based on the methods of trajectories construction using a depth gauge sensor Kinect. The general approach of the work is based on the sequential execution of three phases: obstacle detection, mapping and construction barriers trajectories. Recognition guidelines robot is implemented using an SURF algorithm and FLANN library. Their combined use provides effective recognition obstacle in the near zone of the mobile robot to the required accuracy in real time. Mapping obstacles (concave shells) and the estimate of the distance to the border barriers is based on getting the so-called acceptable points. For this purpose, using the algorithm implemented Pointcloud detection of different types of obstacles in the close range of the robot. Synthesized a modified 3D-pointcloud algorithm which provides a solution to the problem of the robot move without colliding with obstacles. These algorithms can be applied to any stereo cameras and rangefinders that return colored frames and image depth, as described algorithms use data on the distance of objects from the sensor and are resistant to the visible spectrum of light (can operate in total darkness, without requiring a backlight). Built three-level robot control system in buildings and structures on the basis of a modified algorithm of 3D-pointcloud. The results of experiments confirming the effectiveness of the above approach.

**Keywords:** kinect sensor, robot, SURF, 3D-point cloud, image recognition library FLANN, the control system, the trajectory of the robot, the robot collision avoidance, navigation

For citation:

**Nguyen Tuan Dung, Shcherbatov I. A., Protalinskii O. M.** Mobile Robot Control System Based on a Modified 3D-Pointcloud Algorithm, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2016, vol. 17, no. 6, pp. 400–406.

DOI: 10.17587/mau.17.400-406

## References

1. **Hwang Y., Ahuja N.** Gross motion planning — a survey, *ACM Computing Surveys*, 1992, 24 (3), pp. 219–291.
2. **Cunha S., de Matos A., Pereira F.** An automatic path planning system for autonomous robotic vehicles, *Proc. of International Conference on Industrial Electronics, Control, and Instrumentation (IECON '93)*, Maui, HI, USA, 1993, vol. 3, pp. 1442–1447.
3. **Hassouna Sabry M., Alaa E. Abdel-Hakim, Aly A. Farag.** PDE-based robust robotic navigation, *Computer Vision and Image Processing Laboratory (CVIP)*, Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY 40292, USA, 2007, pp. 10–18.
4. **Juan L., Gwon O.** A Comparison of SIFT, PCA-SIFT and SURF, *International Journal of Image Processing (IJIP)*, 2009, vol. 3, no. 4, pp. 143–152.
5. **Bay H., Ess A., Tuytelaars T., Van Gool L.** Speeded-Up Robust Features (SURF), *Computer Vision and Image Understanding*, June 2008, vol. 110, iss. 3, pp. 346–359.
6. **Ramisa A., Vasudevan S., Aldavert D., Toledo R., Lopez de Mantaras R.** Evaluation of the SIFT Object Recognition Method in Mobile Robots, available at: [http://www.iiaa.csic.es/~mantaras/sift\\_eval.pdf](http://www.iiaa.csic.es/~mantaras/sift_eval.pdf).
7. **Noskov V. P., Noskov A. V.** Navigatsiya mobil'nykh robotov po dal'nometricheskim izobrazheniyam (Mobile Robot's Navigation by Means of 3-D Laser Images), *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2005, no. 12, pp. 16–21 (in Russian).
8. **Mojtahedzadeh R.** Robot Obstacle Avoidance using the Kinect, Master of Science Thesis Stockholm, Sweden, 2011, 61 p.
9. **Murphy R.** Introduction to AI Robotics, MIT press, 2000, 400 p.
10. **Bohlin R., Kavradi L.** Path planning using lazy prm / R. Bohlin, *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 1997, pp. 333–338.
11. **Rao N. S. V., Hareti S., Shi W., Iyengar S.** Robot navigation in unknown terrains: introductory survey of non-heuristic algorithms, Oak Ridge National Laboratory, Technical Report ORNL/TM-12410, 1993.
12. **Hoff K., Culver T., Keyser J., Lin M., Manocha D.** Interactive motion planning using hardware-accelerated computation of generalized Voronoi diagrams, *Proc. of the IEEE International Conference on Robotics and Automation*, 2000, pp. 2931–2937.
13. **Zhu D., Latombe J.** New heuristic algorithms for efficient hierarchical path planning, *IEEE Transaction on Robotics and Automation*, 1991, 1 (1), pp. 9–20.
14. **Hwang Y. K., Ahuja N.** Potential field approach to path planning, *IEEE Transaction on Robotics and Automation*, 1992, 8 (1), pp. 23–32.
15. **Navneet Dalal, Bill Triggs.** Object Detection using Histograms of Oriented Gradients, available at: <http://www.webcitation.org/6DvoEuAvL>.
16. **Nguyen Tuan Dung, Shcherbatov I. A.** Raspoznavanie ob'ektov v sisteme tekhnicheskogo zreniya mobil'nogo robota: ispol'zovanie biblioteki FLANN i algoritma SURF (Object recognition in machine vision system of mobile robot: the using of library FLANN and algorithm SURF), *Prikladnyy Zhurnal Upravlenie i Vysokie Tehnologii*, 2014, no. 4, pp. 65–76 (in Russian).
17. **Nguyen Tuan Dung, Shcherbatov I. A.** Sovmestnoe raspoznavanie podvizhnykh i nepodvizhnykh ob'ektov v sisteme tekhnicheskogo zreniya robota (Joint Recognition of the Moving and Stationary Objects in the Machine Vision Systems of Robots), *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2015, vol. 16, no. 7, pp. 464–470 (in Russian).
18. **Shcherbatov I. A., Nguyen Tuan Dung.** Speed up algorithms for obstacles detection in the robot vision system using point cloud, *Creativity in Intelligent Technologies & Data Science*, Volgograd State Technical University, Russia, Springer, September 15–17, 2015, pp. 673–683. DOI: 10.1007/978-3-319-23766-4\_53.
19. **Chakraborty S., Nagwani N. K., Dey L.** Performance Comparison of Incremental K-means and Incremental DBSCAN Algorithms, *International Journal of Computer Applications* (0975 — 8887), August 2011, vol. 27, no. 11, pp. 14–18.
20. **Rassel S. Dzh., Norvig P.** *Iskusstvennyy intellekt: sovremennyy podkhod* (Artificial Intelligence: A Modern Approach), Trans. from English. and ed. K. A. Ptitsyna, Moscow, Vil'jams, 2006, 1408 p. (in Russian).
21. **Protalinskij I. O., Elizarov D. V., Kirilin S. A.** *Universal'naya mobil'naya platform dlya robotov, obsluzhivayushchikh sotsial'nuyu i bytovuyu sfery* (Universal mobile robotic platform for serving the social and domestic sphere), *Vestnik Astrahanskogo Gosudarstvennogo Tekhnicheskogo Universiteta. Seriya: Upravlenie, Vychislitel'naya Tekhnika i Informatika*, 2011, no. 2, pp. 49–54 (in Russian).