

Алгоритмы ускоренной обработки изображений препятствий в системе технического зрения робота

Туан Зунг Нгуен

Федеральное государственное образовательное учреждение высшего профессионального образования «Астраханский государственный технический университет», г. Астрахань

Аннотация: Совместное обнаружение препятствий в ближней зоне мобильного робота является крайне актуальной и сложной задачей. В общем случае окружающая среда робота очень сложна из-за различий в изменении условий освещения и углов зрения. Эти проблемы приводят к снижению точности распознавания препятствий. Различия могут измениться быстро. Но препятствия должны занимать определенную область в пространстве. В работе предложены алгоритмы обнаружения препятствий с алгоритмом «3д-облака точек». Алгоритм содержит основные шаги: Создание 3D-облака точек, преобразование «2D- PointCloud», получение результатов. Препятствия обнаружатся на основе своих определенных областей. Использование 3D-облака точек позволяет удовлетворить требованиям по обнаружению препятствий (подвижных и неподвижных объектов) в режиме реального времени. Результаты экспериментов показывают эффективность предложенного подхода при использовании в составе системы технического зрения мобильной робототехнической платформы.

Ключевые слова: 3д-облако точек, 2д-облако точек, препятствие, мобильный робот, распознавание изображений, изображения глубины, алгоритм понижения разрешения, алгоритм кластеризации, k-mean clustering, плотностный алгоритм кластеризации пространственных данных с присутствием шума.

Введение

Роботизированные мобильные комплексы различного назначения [1] должны иметь возможность автономного исследования окружающей среды (например, в заранее неизвестных условиях зданий и сооружений). Поэтому для целей управления [2] роботу в режиме реального времени необходимо обеспечивать решение задачи объезда различных типов препятствий для перемещения без столкновений.

В настоящее время разработано большое количество методов, алгоритмов и систем навигации роботов. В общем случае существенными являются три аспекта – текущее местоположение, место назначения и траектория движения в процессе перемещения. Для обеспечения перемещения мобильного робота из исходной точки в целевую применяются

различные методы определения расстояния до препятствия и его размеров [3, 4], системы компьютерного зрения [5] с использованием видео и стереокамер [6], датчиков Kinect [7 - 9] и пр.

Для целей определения текущего местоположения мобильного робота в пространстве наиболее перспективной является технология обнаружения с применением Pointcloud и Microsoft Kinect [10], реализованные с использованием библиотеки OPENNI и Pointcloud PLC. Задача уменьшения времени обнаружения препятствий является крайне важной. Для ее решения применяются способы, большинство из которых базируется на обработке RGB-изображений, а значит, имеет место ряд характеристик изображения, оказывающих существенное влияние на качество распознавания. К ним относятся: изменение условий освещения; угол зрения камеры, установленной на роботе. Для устранения влияния этих параметров могут применяться данные глубин изображения, получаемые датчиком Kinect, т.к. они являются устойчивыми к изменению цвета, освещения и простыми с точки зрения отображения [11].

Постановка задачи

При обнаружении препятствий появляется ряд трудностей: масштаб, место, фон и помехи, проекция, вращение и угол обзора. Для их устранения необходимо разработать алгоритм обнаружения, использующий информацию о глубинах изображений, например, от датчика Kinect. Для этого требуется решить две взаимосвязанные подзадачи:

- реализовать алгоритм Pointcloud в среде Visual C#
- исследовать возможности алгоритма Pointcloud для обнаружения любых препятствий в поле зрения робота;
- предложить способ ускорения данного алгоритма и увеличения качества обнаружения объектов.

Общий алгоритм работы системы

Применение видеокамер, т.е. обнаружения объектов в потоке видео, является вычислительно сложным процессом и не обеспечивающим требуемую точность из-за изменения уровня освещенности и угла обзора [7, 12]. При отсутствии света использование камер также затруднительно. В последнее время в данной области произошел существенный сдвиг, в связи с появлением инфракрасных сенсоров глубины, таких как Kinect [7]. Kinect является новой системой камер. Его датчик глубины состоит из инфракрасного проектора, объединенного с монохромной КМОП-матрицей, что позволяет ему получать трёхмерное изображение при любом естественном освещении [7]. Основа работы системы распознавания объектов - алгоритм, показанный на рис. 1. Опишем каждый из блоков, входящих в общий алгоритм работы системы.

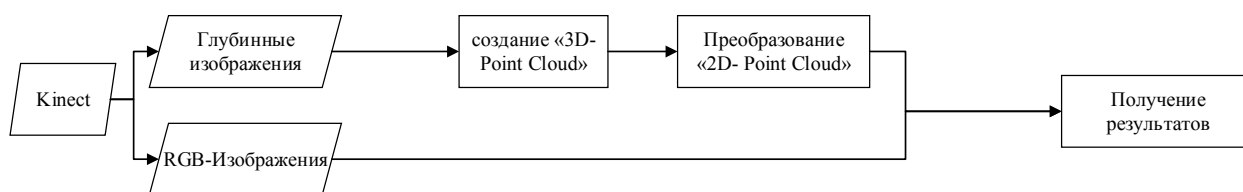


Рис. 1. Общая схема системы обнаружения препятствий с алгоритмом «Pointcloud»

Облако точек (Pointcloud)

Камеры глубины имеют ряд преимуществ перед традиционными датчиками интенсивности, работающими при низком уровне освещенности, позволяя получать калиброванную масштабную оценку, инвариантную по отношению к цвету и текстуре. Получаемые изображения глубины имеют обычное разрешение 640x480 пикселей при 30 кадрах в секунду.

Пусть множество пикселей на изображении глубины $P = \{P_0, \dots, P_i, \dots, P_n\}$, где $n=640*480=307200$. Каждый пиксель P_i описывается тройкой $P_i = \{x_i, y_i, z_i\}$, где x_i - значение ширины пикселя (обычно, $0 \leq x_i \leq 640$), y_i - значение высоты пикселя ($0 \leq y_i \leq 480$), z_i - значение глубины пикселя.

Каждый пиксель P_i соответствует точке T_i в ближней зоне мобильного

робота. Точка T_i описывается тройкой $T_i = \{x_i^t, y_i^t, z_i^t\}$. Для получения пикселя P_i из точки T_i Kinect использует одну матрицу вращения ($MВ$) и один вектор преобразования (ВП). Метод преобразования формулируется как: $P_i = MВ * T_i + ВП$, или

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} mв_{11} & mв_{12} & mв_{13} \\ mв_{21} & mв_{22} & mв_{23} \\ mв_{31} & mв_{32} & mв_{33} \end{bmatrix} \begin{bmatrix} x_i^t \\ y_i^t \\ z_i^t \end{bmatrix} + \begin{bmatrix} вп_x \\ вп_y \\ вп_z \end{bmatrix}$$

Kinect проецирует не только одну точку, но и большое количество точек образца. Он имеет эталонное изображение, которое получается из ИК-камеры, когда все точки поверхности находятся на определенном известном расстоянии. Он находит соответствия между образцами, которые захвачены с камеры и внутреннего эталонного изображения. При сравнении горизонтального положения точки в захваченном и эталонном изображениях может быть получено горизонтальное значение дельты. На самом деле Kinect не рассчитывает глубину, он возвращает более абстрактное значение для обработки. В то время как Kinect SDK может использовать это значение для получения 11-битных значений глубины.

Когда изображение глубины получено с Kinect SDK, становится возможным получение облака точек. Облако точек обычно определяется как набор неорганизованных иррегулярных точек в 3D $P = \{p_1, p_2, \dots, p_i, \dots, p_k\}$, где $k \leq n$. Каждая точка p_i описывается $p_i = \{x_i, y_i, z_i, in_i\}$, где $\{x_i, y_i, z_i\}$ является координатой точки; in_i - дополнительная информация о точке (например, RGB цвета, значения интенсивности и т.д.).

Создание 3D-облака точек является ключевым этапом (рис. 2).

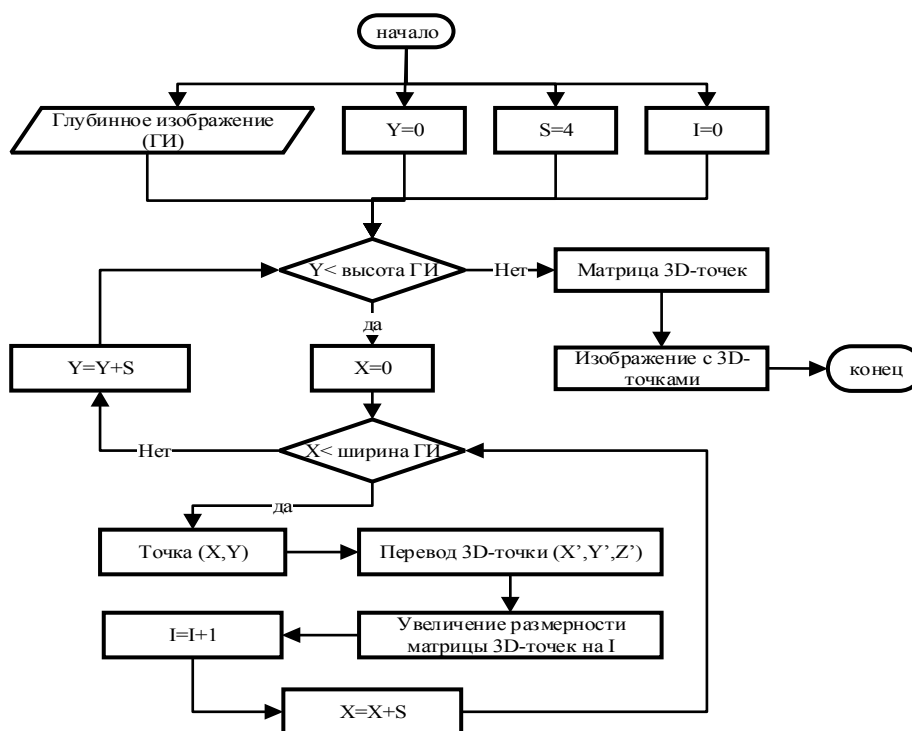


Рис. 2. Схема алгоритма создания «3D- PointCloud»

Метод получения местоположения препятствий в кадре может использовать информацию, хранящуюся в массиве глубин. Каждый пиксель в этом массиве кодирует всю информацию о координатах объектов. Матрица 3D-точек (MaI) получится из этого массива.

Для ускорения вычисления по алгоритму используется понижение разрешения (downsampling). Поэтому размер матрицы MaI уменьшится в вычислениях следующего шага. Например, если размеры глубинного изображения являются 320×240 , максимальный размер массива глубин равен 76800 ($320 \times 240 = 76\ 800$), но максимальный размер выходной матрицы MaI является 4 800 ($320 \times 240 / S^2$; $S=4$). Результаты метода создания 3D-облака точек показаны на рис.3.

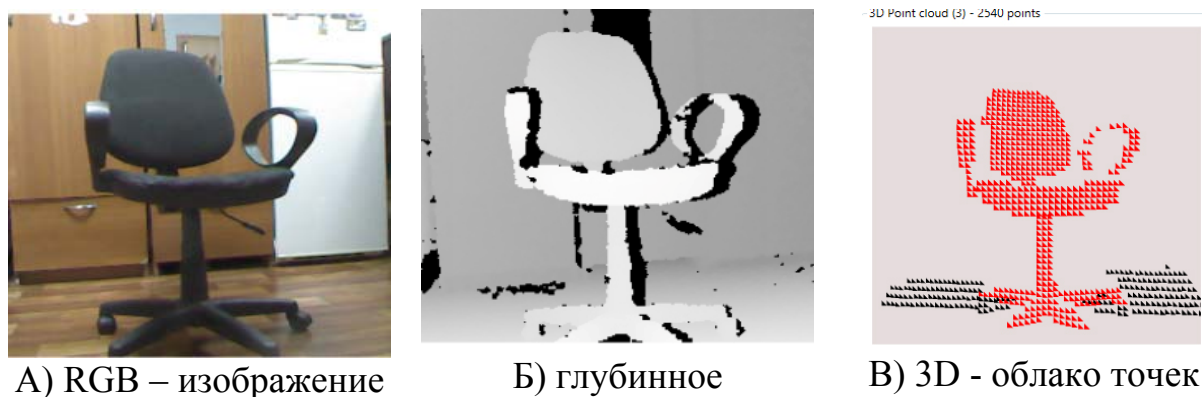


Рис. 3. Получение «3D- PointCloud» с датчика Kinect
Преобразование «2D- PointCloud»

Следующим шагом является отображение облако точек в 2-мерном пространстве. Фиксируются высоты точек между минимальной и максимальной высотами робота. Все точки, которые не могут быть объяснены как препятствия будут удалены. В этом шаге матрица 3D-точек $Ma1$ является исходной. А матрица 2D-точек ($Ma2$) является выходной (рис. 4).

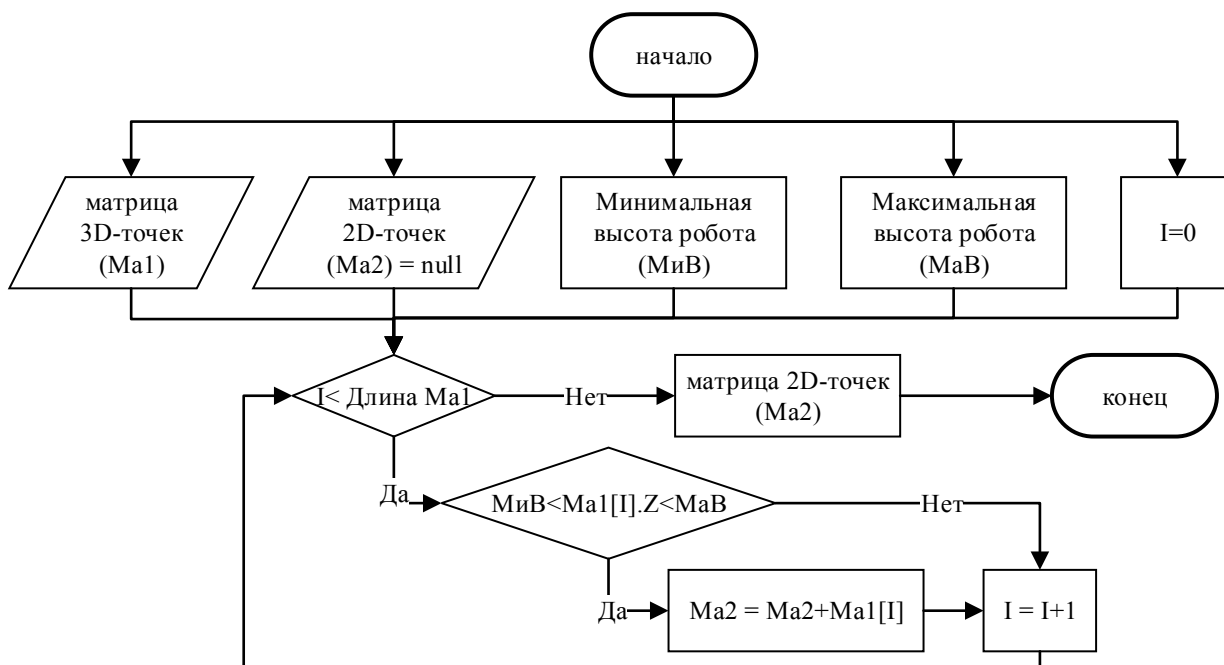


Рис. 4. Схема алгоритма преобразования «2D- PointCloud»

На данном шаге получается матрицу 2D-точек $Ma2$ (рис. 5).



А) 3D - облако точек

Б) 2D - облако точек

Рис. 5. Создание «2D- PointCloud»

Получение результатов

На этом шаге препятствия в поле зрения робота получаются с помощью алгоритма кластеризации. Каждый кластер является препятствием.

Кластеризация является задачей группировки набора объектов в одной и той же группе (кластере), в которой объекты более похожи друг на друга, чем в других группах [13].

Кластерный анализ не является одним конкретным алгоритмом. Популярные представления о кластерах включают: группы с малыми расстояниями между членами кластера; плотные области пространства данных; интервалы или конкретные статистические распределения и пр. Соответствующий алгоритм кластеризации и настройки параметров (в том числе таких как функция расстояния, пороговая плотность или количество ожидаемых кластеров) зависят от индивидуального набора данных и предполагаемого использования результатов. Кластерный анализ является итеративным процессом обнаружения знаний или интерактивной многоцелевой оптимизацией, который включает испытания и неудачи. Необходимо будет модифицировать предварительно обработанные данные и параметры модели для получения результатов с требуемыми свойствами.

В настоящее время для целей обеспечения кластеризации в робототехнике применяются два известных алгоритма: k-mean clustering, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [14].

Классический алгоритм k-mean clustering [15] был представлен

Хартиганом (Hartigan, 1975). Это один из самых простых методов кластеризации данных. Основная операция алгоритма показана на рис 6.

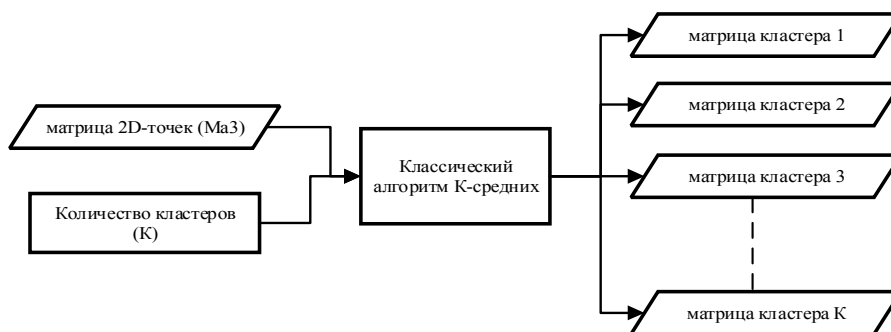


Рис. 6. Иллюстрация операции классического алгоритма «k-meansclustering»

Алгоритм работает очень быстро, если имеется конкретная информация о окружающей среде робота. Но качество обнаружения не очень высокое. Потому что, алгоритм не может удалить шумы.

Поэтому в работе использован алгоритм DBSCAN [16]. Это алгоритм кластеризации на основе плотности. Он находит количество кластеров, начиная с расчетной плотности распределения соответствующих узлов. DBSCAN является одним из наиболее распространенных алгоритмов кластеризации, а также наиболее цитируемых в научной литературе. DBSCAN использует два параметра: ϵ (EPS) и необходимое минимальное количество точек для формирования плотной области (minPts). Работа алгоритма начинается с произвольной начальной точки, извлекается ϵ -окрестность этой точки, а если он содержит достаточно много точек, тогда кластер запускается. В противном случае точка обозначается как шум. Необходимо отметить, что эта точка может быть найдена позже в ϵ -окрестности другой точки и, следовательно, будет включена в состав кластера [16].

Если точка найдена, чтобы быть плотной частью кластера, его ϵ -окрестность также является частью этого кластера. Таким образом, все точки добавляются, которые находятся в пределах ϵ -окрестности. Этот процесс

продолжается до тех пор, пока плотность соединенного кластера найдена полностью. После этого новые не посещенные точки извлечены и обработаны, что приводит к открытию еще кластера или шума. Основная операция алгоритма показана на рис 7.

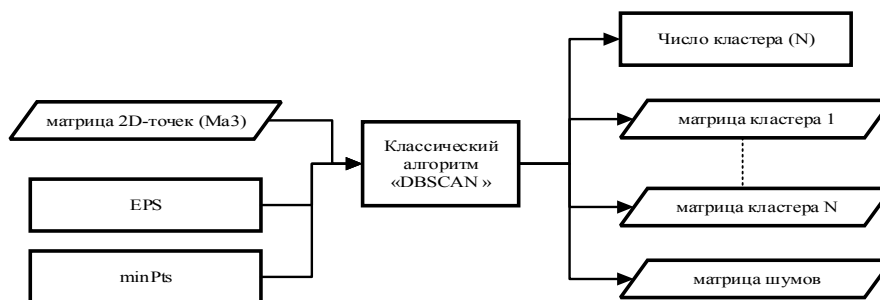


Рис. 7. Иллюстрация операции классического алгоритма «DBSCAN»
Обозначения: «EPS» и «minPts» – основные параметры алгоритма

Алгоритм работает не очень быстро. Но качество обнаружения высокое, т.к. алгоритм может удалить шумы.

Эксперимент и анализ результатов

Для экспериментальной проверки рассматриваемых методов на языке C# (Microsoft Visual C# 2010) с помощью библиотеки EMGUCV разработано программное обеспечение.

Эксперименты проводились на персональной ЭВМ со следующими характеристиками: центральный процессор Intel Pentium Duo T2390 с частотой 1,87ГГц, объем оперативной памяти 1Гб, объем видеопамати 256Мб (ЭВМ с такими характеристиками может устанавливаться на мобильные робототехнические платформы [17, 18]).

Эксперименты проводились для обнаружения препятствий с использованием облака точек, полученные результаты представлены на рис. 8.

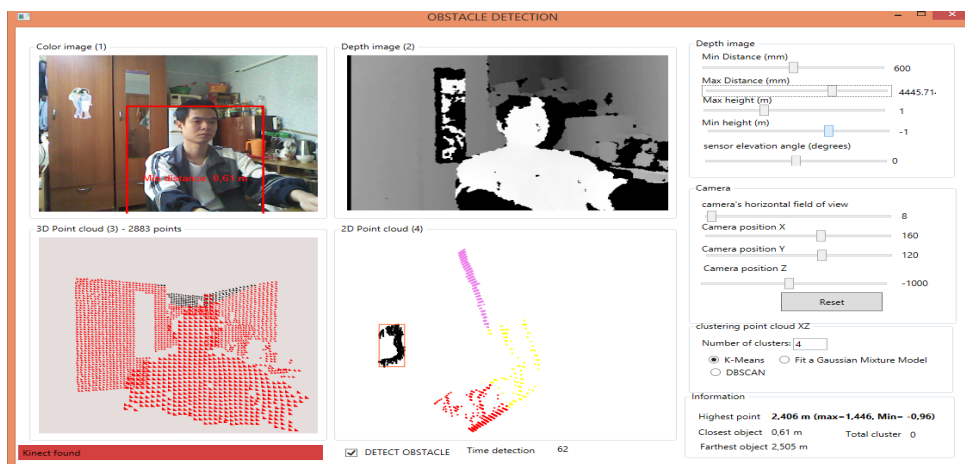


Рис. 8. Интерфейс программы применения 3D-облака точек для обнаружения препятствий

Для обнаружения препятствий обычно используется алгоритм поиска ближайших соседей совместно с алгоритмом SURF (Метод2) [19]. Этот метод удовлетворяет жестким требованиям по обнаружению объектов в режиме реального времени.

Результаты экспериментов сравнения метода 1 и метода 2 представлены на рис. 9 и в табл. 1.

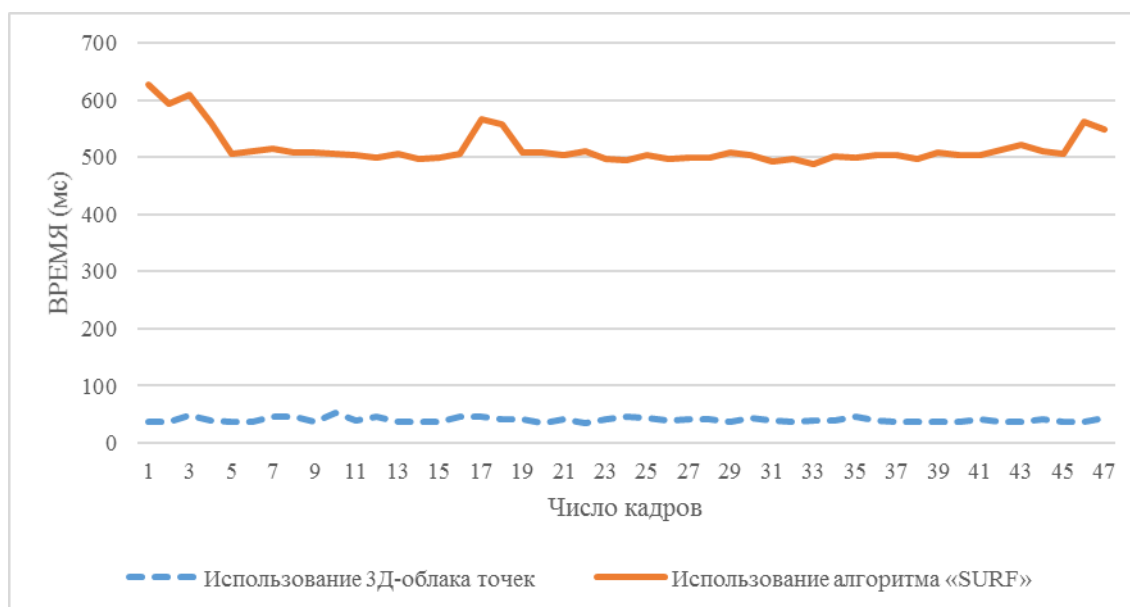


Рис. 9. Быстродействие алгоритма распознавания неподвижных объектов

Таблица № 1

Сравнение с методом применения алгоритма «SURF»

№	Вид эксперимента	Количество входных изображений	Среднее время обнаружения объекта, мс
1	С применением 3D-облака точек (Метод 1)	47	40,38
2	С применением алгоритма SURF (Метод 2)	47	516,17

Из табл.1 видно, что время на обнаружение объекта с использованием первого метода в 13 раз меньше, чем второго метода.

Как показали ранее, появляется ряд трудностей: масштаб, место, фон и помехи, проекция, вращение и угол обзора при обнаружении препятствий. В экспериментах рассмотрено изменение интенсивность освещения к качеству обнаружения в различное время (в 10:00, 14:00, 19:00, 23:00) без искусственного освещения, полученные результаты представлены на рис. 10 и в табл. 2.

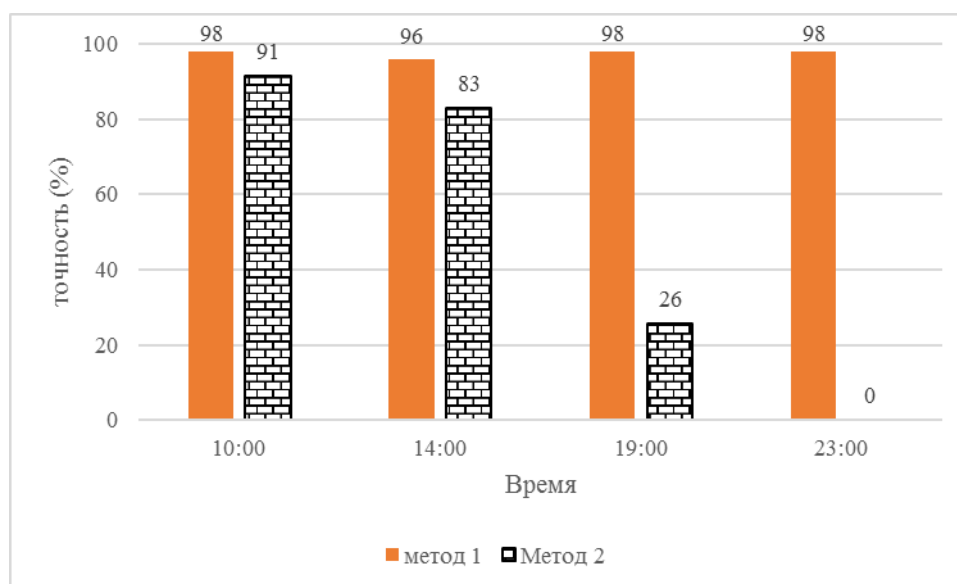


Рис. 10. Сводная диаграмма результатов исследования

Таблица № 2

Сравнительный анализ результатов

Время	Количество входных изображений	Метод 1		Метод 2	
		Ошибка	Точность, %	Ошибка	Точность, %
10:00	47	1	98	4	91
14:00	47	2	96	8	83
19:00	47	1	98	35	26
23:00	47	1	98	47	0

Из табл. 2 видно, что качество обнаружения метода 1 не зависит от интенсивности света. Препятствия успешно обнаруживаются даже когда отсутствует освещение (в 23:00). Потому что Kinect может использовать инфракрасный проектор для получения глубинных изображений.

Результаты экспериментов свидетельствуют о том, что существует возможность применять алгоритм Pointcloud для обнаружения препятствий в системе технического зрения робота. Однако, для увеличения быстродействия традиционного алгоритма, следует применять алгоритм понижения разрешения в шаге «Создание 3D-облака точек» и алгоритм DBSCAN в шаге «Применение алгоритмы кластеризации».

Заключение

Данные экспериментов показывают, что обнаружение любых препятствий следует производить с использованием алгоритма Pointcloud. Эффективность данного алгоритма выше, чем традиционных алгоритмов, которые базируются на изображениях, полученных из видимого спектра по аналогии с тем, как это воспринимается человеческим глазом. При этом применение алгоритма понижения разрешения и DBSCAN существенно увеличивает быстродействие алгоритма распознавания объектов и повышает его точность.

Полученные результаты свидетельствуют о том, что использование облака точек удовлетворяет жестким требованиям по обнаружению любых

препятствий в режиме реального времени. Это означает, что предложенный подход может применяться в системах технического зрения мобильных робототехнических платформ для получения сенсорной карты внешней среды, используемой системой управления роботом для перемещения в зданиях и сооружениях в заранее неизвестной среде.

Литература

1. Голубкин И.А., Кирилин С.А., Щербатов И.А. Принципы разработки мультимедийных прикладных роботизированных устройств. Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика, 2010. № 2. С. 153-157.
2. Щербатов И.А. Интеллектуальное управление робототехническими системами в условиях неопределенности. Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2010. № 1. С. 73-77.
3. Носков, В.П. Навигация мобильных роботов по дальнометрическим изображениям / В.П. Носков, А.В. Носков // Мехатроника, автоматизация, управление, 2005. № 12. С. 16 - 21.
4. Голубкин И.А., Щербатов И.А. Универсальная сенсорная подсистема мобильного колесного робота // Датчики и системы, 2010. № 8. С. 32-35.
5. Forsyth, D.A. Computer Vision: A Modern Approach / D.A. Forsyth, J. Ponce // Robotics and Autonomous Systems, 2006. Vol. 11, № 3. pp. 84 - 92.
6. Bonin-Font, F. Visual Navigation for Mobile Robots: a Survey/ F. Bonin-Font, A. Ortiz, G. Oliver // Journal of Intelligent and Robotic System, 2008. pp. 263 – 296.
7. Kefer, M., Kubinger W. Evaluation of Kinect Depth Sensor for Use in Mobile Robotics / Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium, Volume 22, No. 1. pp. 147-148.
8. Нгуен Туан Зунг, Щербатов И.А. Распознавание объектов в системе технического зрения робота: использование библиотеки FLANN и алгоритма SURF // Прикаспийский журнал: управление и высокие технологии. 2014. № 4 (28). С. 65-76.
9. Нгуен Туан Зунг, Щербатов И.А. Применение Kinect и библиотеки Flann для построения сенсорной карты среды мобильного робота // Фундаментальные и прикладные исследования в современном мире. 2014. Т. 1. № 8. С. 130-135.
10. Mojtahedzadeh R. Robot Obstacle Avoidance using the Kinect. Master of Science Thesis Stockholm, Sweden 2011. 61 p.

11. Shotton J., Fitzgibbon A. Cook M., Sharp T., Finocchio M., Moore R., Kipman A., Blake A. Real-time human pose recognition in parts from single depth images. - VISAPP International Conference on Computer Vision Theory and Applications, February 2013. pp. 119-135.
12. Обзор информации о роботизированных комплексах. URL: mchs.gov.by/_modules/_cfiles/files/obzor_o_robotizirovannih_kompleksah.pdf.
13. Jagadish H.V.. Linear Clustering of Objects with Multiple Attributes. / ACM, 1990. pp. 332-342.
14. Sanjay Chakraborty., N.K.Nagwani., LopamudraDey. Performance Comparison of Incremental K-means and Incremental DBSCAN Algorithms. - International Journal of Computer Applications (0975 – 8887), Volume 27– No.11, August 2011. pp. 14-18.
15. Jain A.K., Murty M.N., Flynn P.J. Data clustering: a review // ACM Computing Surveys. 1999. 31, N 3. pp.264–323.
16. Ester M., Kriegel H.P., Sander J., Xu X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings KDD-96, 1996. pp. 226-231.
17. Protalinskii I.O., Sherbatov I.A., Shishkin N.D. Optimal strategy synthesis for a group of mobile robots with variable structure // World Applied Sciences Journal, 2013. Т. 24. № 24. pp. 268-275.
18. Щербатов И.А., Проталинский И.О., Проталинский О.М. Управление группой роботов: компонентный подход // Информатика и системы управления. 2015. № 1 (43). С. 93-104.
19. Bay H., Ess A., Tuytelaars T., Van Gool L. Speeded-Up Robust Features (SURF). - Computer Vision and Image Understanding, Vol 110, Issue 3, June 2008, pp.346-359.
20. Файзрахманов Р.А., Полевщиков И.С. Моделирование и автоматизация процесса управления формированием профессиональных умений и навыков оператора производственно-технологической системы. // Инженерный вестник Дона, 2014, № 4. URL: ivdon.ru/ru/magazine/archive/N4y2014/2705.
21. Вилесов Л.Д., Кузьмин С.А. Алгоритмы сегментации кадров и сжатия видеoinформации для систем видеонаблюдения. // Инженерный вестник Дона, 2014, № 4. URL: ivdon.ru/ru/magazine/archive/N4y2014/2733.
22. Копылов С.А. Обзор методов и подходов обнаружения отказов и отказоустойчивости системы управления для роботизированных морских подвижных объектов. // Инженерный вестник Дона, 2014, № 3. URL: ivdon.ru/ru/magazine/archive/n3y2014/2509

References

1. Golubkin I.A., Kirilin S.A., Shherbatov I.A. Vestnik Astrahanskogo
-



gosudarstvennogo tehničeskogo universiteta. Serija: Upravlenie, vychislitel'naja tehnika i informatika, 2010. № 2. pp.153-157.

2. Shherbatov I.A. Vestnik Astrahanskogo gosudarstvennogo tehničeskogo universiteta. Serija: Upravlenie, vychislitel'naja tehnika i informatika, 2010. № 1. Pp. 73-77.

3. Noskov, V.P. Mehatronika, avtomatizacija, upravlenie, 2005. № 12. P. 16 - 21.

4. Golubkin I.A., Shherbatov I.A. Datchiki i sistemy, 2010. № 8. p. 32-35.

5. Forsyth, D.A. Computer Vision: A Modern Approach. D.A. Forsyth, J. Ponce. Robotics and Autonomous Systems, 2006. - Vol. 11, № 3. P. 84 - 92.

6. Bonin-Font, F. Visual Navigation for Mobile Robots: a Survey. F. Bonin-Font, A. Ortiz, G. Oliver. Journal of Intelligent and Robotic System, 2008. p. 263 – 296.

7. Kefer, M., Kubinger W. Evaluation of Kinect Depth Sensor for Use in Mobile Robotics. Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium, Volume 22, No. 1. pp. 147-148.

8. Nguyen Tuan Dung, Shherbatov I.A. Prikaspijskij zhurnal: upravlenie i vysokie tehnologii, 2014. № 4 (28). pp. 65-76.

9. Nguyen Tuan Dung, Shherbatov I.A. Fundamental'nye i prikladnye issledovaniya v sovremennom mire, 2014. T. 1. № 8. pp. 130-135.

10. Mojtahedzadeh R. Robot Obstacle Avoidance using the Kinect. Master of Science Thesis Stockholm, Sweden 2011. 61 p.

11. Shotton J., Fitzgibbon A. Cook M., Sharp T., Finocchio M., Moore R., Kipman A., Blake A. Real-time human pose recognition in parts from single depth images. VISAPP International Conference on Computer Vision Theory and Applications, February 2013. pp. 119-135.

12. Obzor informacii o robotizirovannyh kompleksah. Review the information about the robotic complexes. [Review the information about the robotic complexes].URL:mchs.gov.by/_modules/_cfiles/files/obzor_o_robotizirovannyh_k

ompleksah.pdf .

13. Jagadish H.V. Linear Clustering of Objects with Multiple Attributes. ACM, 1990. pp. 332-342.

14. Sanjay Chakraborty., N.K.Nagwani., LopamudraDey. Performance Comparison of Incremental K-means and Incremental DBSCAN Algorithms. International Journal of Computer Applications (0975 – 8887), Volume 27– No.11, August 2011, pp. 14-18.

15. Jain A.K., Murty M.N., Flynn P.J. Data clustering: a review. ACM Computing Surveys, 1999. 31, N 3. pp.264–323.

16. Ester M., Kriegel H.P., Sander J., Xu X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings KDD-96. 1996. pp. 226-231.

17. Protalinskii I.O., Sherbatov I.A., Shishkin N.D. Optimal strategy synthesis for a group of mobile robots with variable structure. World Applied Sciences Journal. 2013. T. 24. № 24. pp. 268-275.

18. Shherbatov I.A., Protalinskij I.O., Protalinskij O.M. Informatika i sistemy upravlenija. 2015. № 1 (43). pp. 93-104.

19. Bay H., Ess A., TuytelaarsT., Van GoolL. Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding, Vol 110, Issue 3, June 2008, pp.346-359.

20. Fajzrahmanov R.A., Polevshhikov I.S. Inženernyj vestnik Dona (Rus), 2014, № 4. URL: ivdon.ru/ru/magazine/archive/N4y2014/2705.

21. Vilesov L.D., Kuz'min S.A. Inženernyj vestnik Dona (Rus), 2014, № 4. URL: ivdon.ru/ru/magazine/archive/N4y2014/2733.

22. Kopylov S.A. Inženernyj vestnik Dona (Rus), 2014, № 3. URL: ivdon.ru/ru/magazine/archive/n3y2014/2509