
ТЕОРЕТИЧЕСКИЕ ОСНОВЫ, РАЗРАБОТКА И ПРИМЕНЕНИЕ МАТЕМАТИЧЕСКИХ МОДЕЛЕЙ, АЛГОРИТМОВ, ПРОГРАММ ДЛЯ ЭВМ

УДК 004.45-004.932: 004.896

РАСПОЗНАВАНИЕ ОБЪЕКТОВ В СИСТЕМЕ ТЕХНИЧЕСКОГО ЗРЕНИЯ МОБИЛЬНОГО РОБОТА: ИСПОЛЬЗОВАНИЕ БИБЛИОТЕКИ FLANN И АЛГОРИТМА SURF

Статья поступила редакцию 23.10.2014, в окончательном варианте 03.12.2014.

Нгуен Туан Зунг, аспирант, Астраханский государственный технический университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 16, e-mail: dunghvkt@yahoo.com

Щербатов Иван Анатольевич, кандидат технических наук, доцент, Астраханский государственный технический университет, 414056, Российская Федерация, г. Астрахань, ул. Татищева, 16, e-mail: sherbatov2004@mail.ru

Обнаружение объектов (препятствий) в ближней зоне мобильного робота является крайне актуальной и сложной задачей. В работе предложено применение для нахождения соответствия изображений в коллекции образов алгоритма SURF (Speeded Up Robust Features) и метода поиска ближайшего соседа в библиотеке «FLANN – Fast Library for Approximate Nearest Neighbors». Библиотека «FLANN» содержит коллекцию алгоритмов: извлечение дескрипторов изображения, вычислительные дескрипторы для множественных изображений; создание индекса FLANN; выполнение поиска KNN (метод k ближайших соседей); получение итоговых результатов. Эффективность нахождения объектов будет увеличиваться в случае применения алгоритма SURF. Использование библиотеки FLANN совместно с алгоритмом SURF позволяет удовлетворить требованиям по обнаружению объектов в режиме реального времени. Результаты экспериментов показывают эффективность предложенного подхода при использовании в составе системы технического зрения мобильной робототехнической платформы.

Ключевые слова: библиотека FLANN, алгоритм SURF, SIFT, мобильный робот, распознавание изображений, K-мерное дерево, поиск ближайшего соседа

OBJECT RECOGNITION IN MACHINE VISION SYSTEM OF MOBILE ROBOT: THE USING OF LIBRARY “FLANN” AND ALGORITHM “SURF”

Nguyen Tuan Zung, post-graduate student, Astrakhan State Technical University, 16 Tatishchev St., Astrakhan, 414056, Russian Federation, e-mail: dunghvkt@yahoo.com

Shcherbatov Ivan A., Ph.D. (Engineering), Associate Professor, Astrakhan State Technical University, 16, Tatishchev St., Astrakhan, 414056, Russian Federation, e-mail: sherbatov2004@mail.ru

Object detection (obstacles detection) in the near field of mobile robot is very urgent and difficult task. In this paper, we propose a method, which uses nearest neighbor search «FLANN – fast library for approximate nearest neighbors», and algorithm «SURF» to find matching images in the collection of images. Library «FLANN» contains a collection of algorithms: extracting descriptors of the image, computing descriptors for multiple images, create index FLANN, performing searches KNN (k-nearest neighbor method), and getting results. Efficiency of finding objects will increase with the use of the algorithm SURF. Using library FLANN conjunction with SURF algorithm meets the requirements for detection of objects in real time. The experimental results show the effectiveness of the proposed approach when used in the vision system of mobile robot platform.

Keywords: the FLANN library, SURF algorithm, SIFT, mobile robot, image recognition, K -dimensional tree, the nearest neighbor search

Введение. Современное состояние развития робототехнических устройств и систем позволяет утверждать, что их применение внутри зданий и сооружений (ЗиС) будет возрастать с течением времени. Сферы применения мобильных роботов в ЗиС – роботы специального назначения [2], рекламно-информационные [3] и пр. При этом одним из ключевых требований к современным мобильным робототехническим платформам является возможность автономной навигации в реальной среде (в том числе в ЗиС) на основе системы технического зрения [1].

Целью настоящей работы является анализ возможностей и вычислительной эффективности использования библиотеки FLANN и алгоритма SURF в системах технического зрения роботов.

Общая характеристика проблематики технического зрения роботов. Существуют системы навигации, использующие информацию о положениях окружающих объектов на основе дальномеров [2,5], видеокамер [1], стереокамер [8] и ряд других. Кроме того, весьма перспективным направлением является разработка систем навигации мобильных роботов в ЗиС на основе инфракрасного сенсора глубины, работающего совместно с видеокамерой – например, Microsoft Kinect [15] или Asus Xtion [10]. Обнаружение объектов в ближней зоне робота обычно обеспечивается применением двух алгоритмов: SURF (Speeded Up Robust Features) [9] и SIFT (Scale Invariant Feature Transform) [18]. Сравнение результатов использования этих двух алгоритмов [14] показывает преимущество SURF [9] над SIFT и его более высокую эффективность. Кроме того, SURF реализован в большинстве современных стандартных библиотек по распознаванию изображений [14].

В робототехнике крайне сложной и актуальной является проблема распознавания объектов в составе системы технического зрения и сенсорного оучувствления. Решение этой задачи зачастую обеспечивается путем поиска соответствия между полученным с камеры изображением и эталонным объектом в коллекции изображений робота. Для решения этой задачи обычно применяются два алгоритма: линейный поиск и приближенный (нечеткий) поиск ближайших соседей [17].

Линейный поиск характеризуется высоким уровнем затрат вычислительных ресурсов и не всегда позволяет обеспечить требуемое быстродействие для применения в режиме реального времени. Для пространств большой размерности отсутствуют эффективные алгоритмы его реализации. Поэтому метод поиска ближайших соседей является предпочтительным по сравнению с простым линейным поиском [17]. Приближенные алгоритмы, к которым относится поиск ближайших соседей, являются более быстродействующими, чем линейный поиск, при этом они обеспечивают не уступающую ему точность.

В настоящее время для целей обеспечения ориентирования робота в пространстве применяются три известных алгоритма: FLANN (Fast Library for Approximate Nearest Neighbors), ANN (A Library for Approximate Nearest Neighbor Searching), STANN (Thread-safe Approximate Nearest Neighbor). При этом FLANN реализован в большинстве библиотек, таких как OPENCV и OPENNI [12].

Постановка задачи. Алгоритм поиска ближайших соседей (Nearest Neighbor Searching, NNS), больше известен как поиск близости, поиск по сходству или поиск ближайшей точки. Близость обычно выражается в терминах функции несходства. Задача поиска ближайших соседей формулируется следующим образом: задано множество S точек в пространстве M и точка запроса $q \in M$, найти ближайшую точку в S до q [16]. Donald Knuth (1973) назвал данную проблему «проблемой почты». Эта проблема является « k -НН поиском», в которой необходимо найти K ближайшие точки.

Обычно M является метрическим пространством [17], выражаемом в метрическом состоянии, которое симметрично и удовлетворяет неравенству треугольника. Чаще M представляется G -мерным векторным пространством, в котором несходство измеряется с помощью евклидова расстояния между объектами (функция несходства может быть произвольной) [16].

Качество и эффективность алгоритмов NSS определяются сложностью запросов и любых поисковых структур данных, которые должны быть обработаны. Однако не существует точных решений для NNS в Евклидовом пространстве большой размерности (при использовании полинома предварительной обработки и полилогарифмического времени поиска) [17].

Для практической реализации NNS существуют многочисленные варианты, которые называются K -мерным деревом: диаграммы Вороного, BSP-деревья, деревья покрытий, VP-деревья, R-деревья. Наиболее известным является K -мерное дерево [16].

Таким образом, для решения задачи обнаружения объектов в ближней зоне робота предлагается использовать библиотеку FLANN совместно с алгоритмом SURF. Тогда данная задача может быть разделена на две подзадачи:

- оценка применимости библиотеки FLANN для нахождения соответствия изображения в коллекции образов, имеющейся в составе системы технического зрения мобильного робота;
- реализация алгоритма SURF.

Методы исследования

K-мерное дерево:

K - d дерево (k -dimensional tree) является структурой данных, разбивающей k -мерное пространство на ряд точек. K - d дерево – бинарное дерево, в котором каждый узел является k -мерной точкой. Каждый не конечный узел может рассматриваться как неявно генерированное расщепление гиперплоскости, которая делит пространство на две части, известные как полупространства. Точки слева от этой гиперплоскости представлены левым поддеревом узла, а точки справа – правым поддеревом.

Структура k - d дерева, описанная на языке C#:

```
Const int K = 10; // количество пространств ключей
Public class Node // структура элемента
{
    Public int[] Keys=new int[K]; // массив ключей элемента
    Public string Infor; // информация об элементе
}
Public class Tree // структура узла дерева
{
    public Node n; // элемент
    public Tree lefttree; // левое поддерево
    public Tree righttree; // правое поддерево
}
```

В настоящее время применяются два типа деревьев – неоднородные и однородные k - d деревья. В однородных k - d деревьях каждый узел хранит запись, а в неоднородных внутренние узлы хранят только ключи, так как листья хранят ссылки на записи [16]. При сокращении K - d дерева до k -мерного оно преобразуется в неоднородную древовидную структуру поиска для сохранения точек из R^k [16]. Последнее обеспечивает схожую возможность поиска на R -дереве в заданном диапазоне.

Точка « t » находится в неоднородном k - d дереве $H_i(t) = (x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_k)$ при $1 \leq i \leq k$ параллельно оси $(k-1)$ -мерной гиперплоскости. Для поиска корня [16] необходимо разделить точки гиперплоскости $H_i(t)$ на два (по возможности одинаково больших) множества точек и записать « t » в корень. Тогда слева сохраняются все точки, у которых $x_i < t$, а справа те, у которых $x_i > t$. Для левого поддерева необходимо разделить точки на новую

«разделенную плоскость» $H_2(t)$, а « t » сохранить во внутреннем узле. Слева от него сохраняются все точки, у которых $x_2 < t$. Этот процесс рекурсивно повторяется для всех пространств. Затем, начиная с первого пространства, повторяется до тех пор, пока каждую точку можно будет идентифицировать через гиперплоскость [16]. Пример k-d дерева показан на рис. 1 (точки различных множеств имеют отличающуюся штриховку).

K-d дерево можно построить за $O(n(k+\log(n)))$ шагов. Поиск диапазона может быть выполнен за $O(n(\exp(1 - (1/k)) + a))$ шагов, при этом « a » обозначает размер результата. Требование к памяти для самого дерева ограничено $O(kn)$ [16].

Структура дерева может меняться в зависимости от деталей реализации алгоритма [2].

Поиск ближайшего соседа:

При поиске ближайших соседей (KNN) алгоритм стремится найти точку в дереве, которая расположена наиболее близко к заданной входной точке [16]. Этот поиск может быть эффективнее при использовании дерева свойств для сокращения пространства поиска (рис. 2).

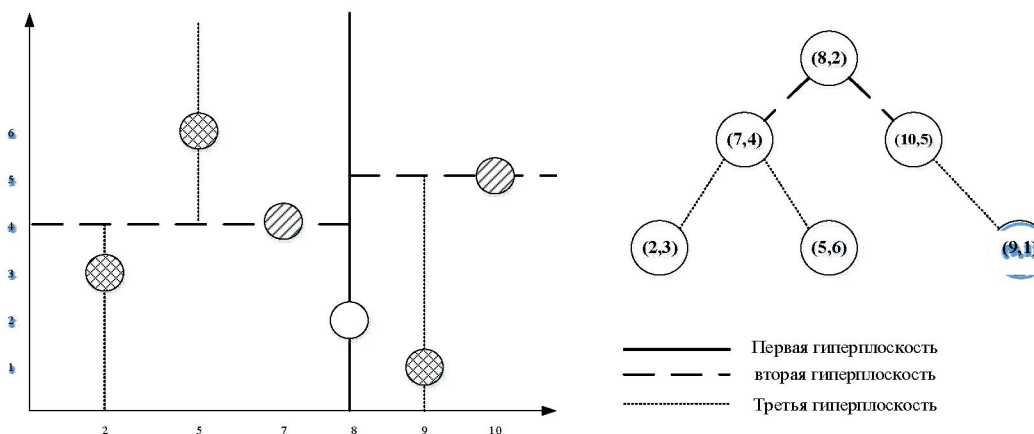


Рис. 1. Пример k-d дерева

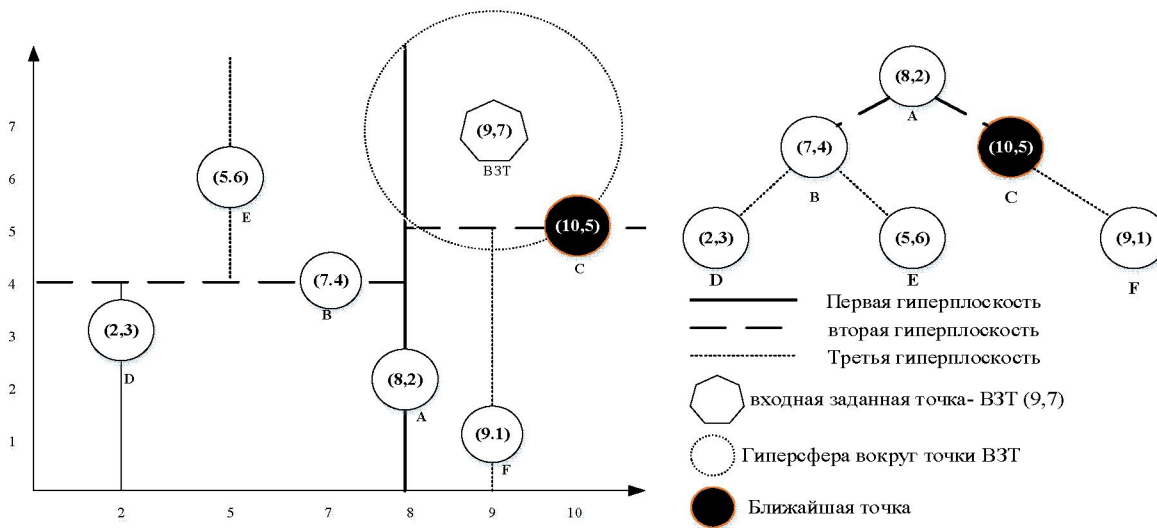


Рис. 2. Пример реализации поиска ближайшего соседа

Алгоритм включает две подзадачи: поиск ближайших элементов в заданном диапазоне и определение возможного ближайшего элемента [16]:

- начиная с корневого узла, алгоритм переходит вниз по рекурсивному дереву влево или вправо в зависимости от того, точка ближе или дальше, чем в текущем узле в месте ветвления (в K -мерном дереве каждый узел или точка имеет свои координаты);
- после того, как алгоритм достигает листового узла, он сохраняет эту точку узла как «текущий лучший вариант»;
- алгоритм «разматывает» рекурсию дерева, выполнив некоторую последовательность шагов на каждом узле;
- когда алгоритм заканчивает этот процесс для корневого узла, то поиск завершается.

При этом достигнутый «текущий лучший вариант» считается оптимальным для условий задачи, представленных деревом.

Алгоритм использует квадраты расстояний для сравнения, чтобы избежать вычисления квадратного корня (так как существует возможность получения отрицательного значения расстояний). Поиск ближайшей точки является в отношении количества действий операцией $O(\log N)$ в случае хаотически распределенных точек. Для увеличения быстродействия алгоритма применяется аппроксимация. Например, алгоритм «best-bin-first search» является самой популярной реализацией задачи поиска ближайшего соседа [17].

Применение библиотеки FLANN:

Одним из способов поиска соответствия изображения в коллекции образов является извлечение всех дескрипторов изображений в коллекции и их обработка. Однако для больших коллекций данный способ малоэффективен. Для устранения этого недостатка используется библиотека FLANN (библиотека быстрого приблизительного поиска ближайших соседей), которая разработана специально для больших наборов данных и содержит коллекцию алгоритмов. Применение FLANN показано на рис. 3.

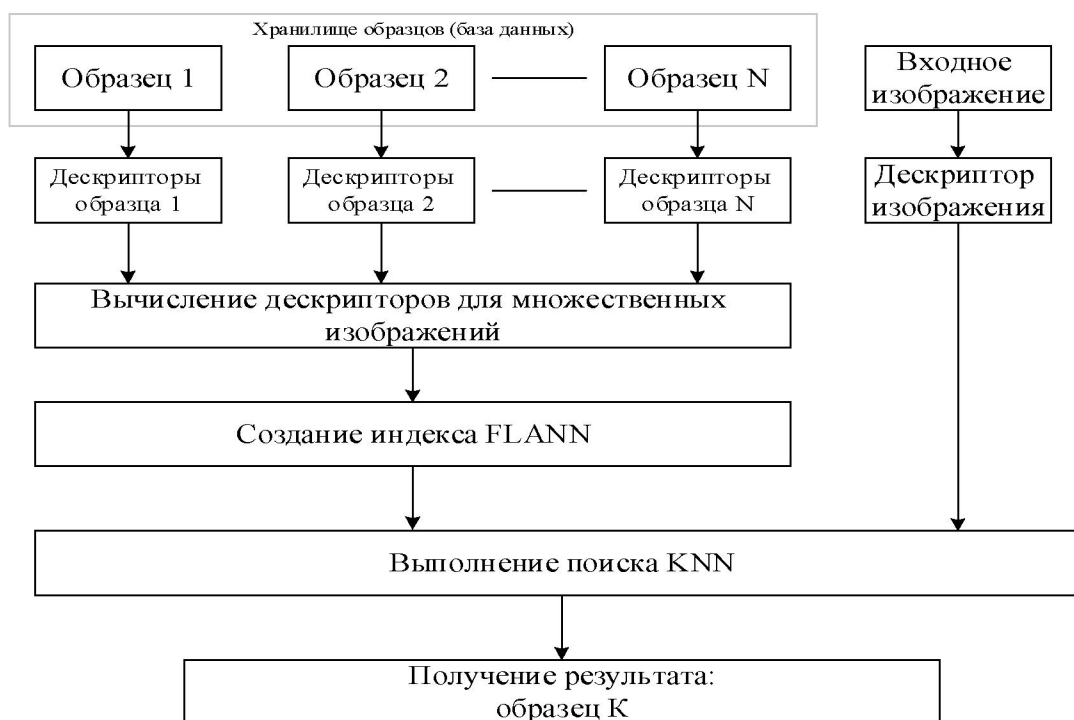


Рис. 3. Общая схема применения библиотеки FLANN

Основные шаги (этапы) использования FLANN: извлечение дескрипторов изображения; вычисление дескрипторов для множественных изображений; создание индекса FLANN; выполнение поиска KNN; получение итогового результата.

Извлечение дескрипторов изображения является ключевым этапом. Для вычисления дескрипторов каждого изображения в коллекции применяется алгоритм SURF (рис. 4).

Вычисление дескрипторов для множественных изображений объединяет все дескрипторы в одну матрицу и отслеживает любой диапазон дескрипторов внутри матрицы, которые получены из каждого образа в базе данных. Эти операции показаны на рис. 5.

Создание индекса FLANN соединённой матрицы [7].

Выполнение поиска KNN: запускается поиск KNN по индексу FLANN. Поиск K -ближайших соседей вычисляет расстояние между дескрипторами входного изображения и всех изображений в коллекции и возвращает K пар с наименьшим расстоянием до классифицируемого объекта.

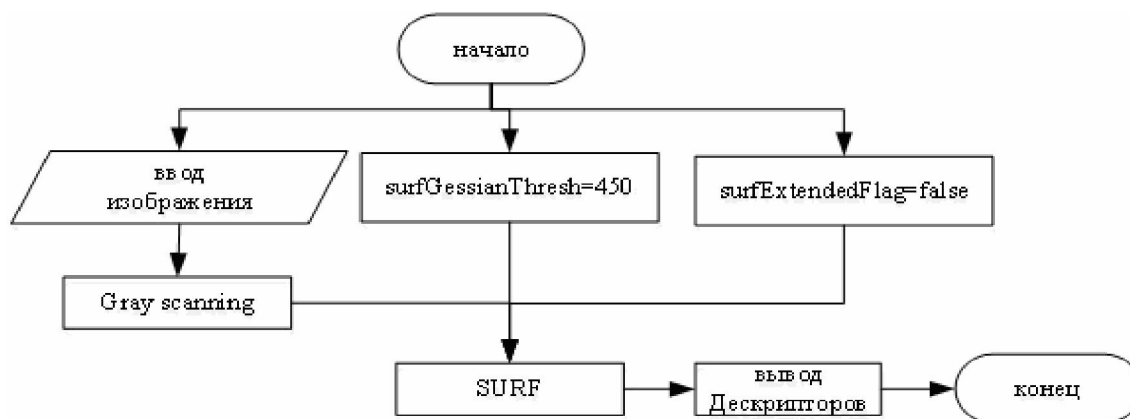


Рис. 4. Извлечение дескрипторов изображения с применением SURF:

Gray scanning – получение серого изображения; surfGessianThresh и surfExtendedFlag – основные параметры алгоритма

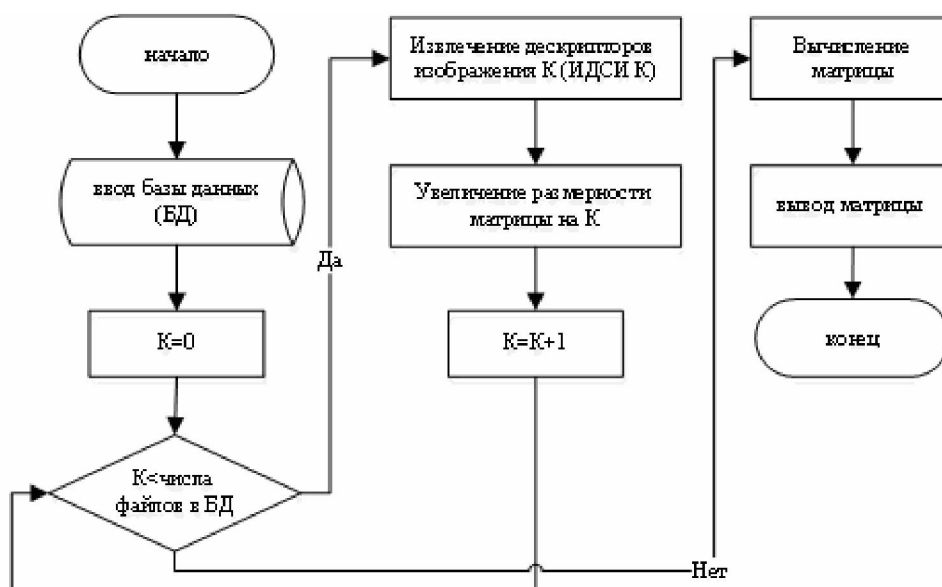


Рис. 5. Алгоритм вычисления дескрипторов

Поиск начинается с верхнего уровня. Для этого определяем вектор v_2 , ближайший к искомому v_1 . Затем рассматривается компонента связности, для которой вектор v_2 является центром [4]. Если количество векторов в этой компоненте связности больше искомого n , то можно рассматривать только эту компоненту. Если меньше n , то производится поиск следующей ближайшей компоненты связности. Далее от центра производится спуск к векторам компоненты следующего уровня и так до тех пор, пока не будет осуществлен спуск до нижнего уровня [4].

Получение результатов (рис. 6): происходит выбор изображения в коллекции образов, которое имеет наибольшее количество совпадающих дескрипторов в матрице индексов (эта матрица получится из шага «Вычисление дескрипторов для множественных изображений»).

Эксперимент и анализ результатов. Для экспериментальной проверки рассматриваемых методов авторами на языке C# (Microsoft Visual C# 2010) с помощью библиотеки EMGUCV было разработано программное обеспечение, которое использует алгоритм SURF и библиотеку FLANN. Кроме того, в функционал разработанного программного обеспечения была включена возможность создания коллекции изображений (табл. 1).

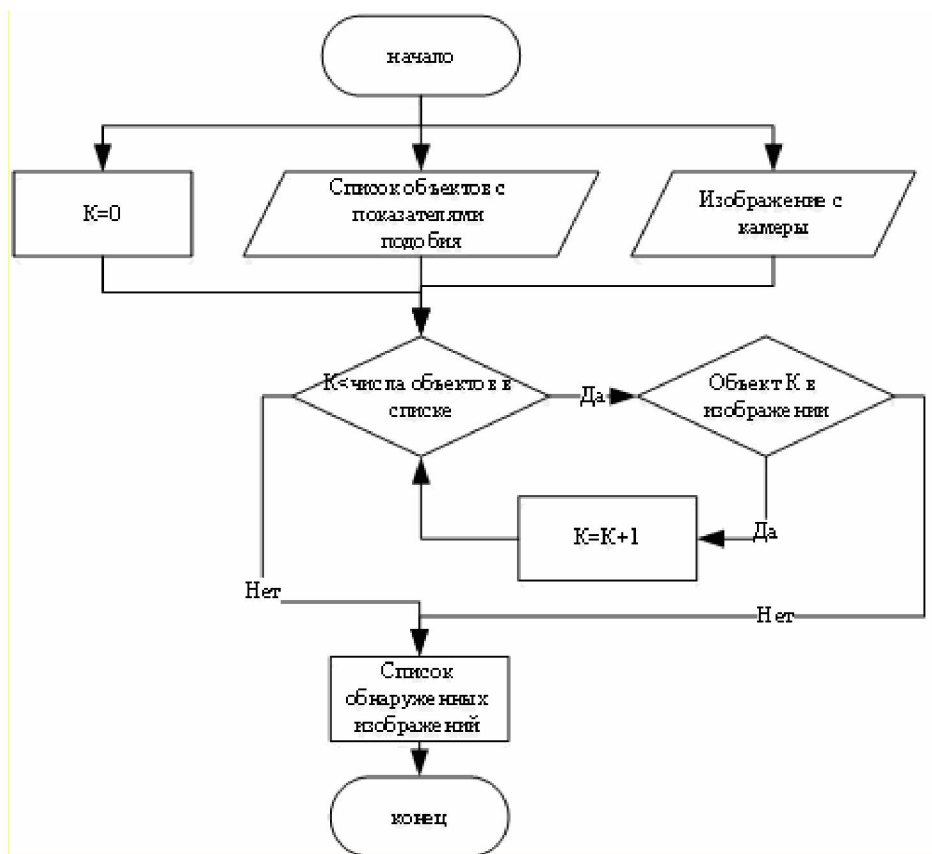


Рис. 6. Схема получения итогового результата

Таблица 1

Пример коллекции изображений, представленной в виде базы данных

 <p>Размеры: 222x425 пикселей Глубина цвета: 24</p>	 <p>Размеры: 234x364 пикселей Глубина цвета: 24</p>	 <p>Размеры: 335x263 пикселей Глубина цвета: 24</p>	 <p>Размеры: 179x201 пикселей Глубина цвета: 24</p>	 <p>Размеры: 220x190 пикселей Глубина цвета: 24</p>
 <p>Размеры: 173x376 пикселей Глубина цвета: 24</p>	 <p>Размеры: 115x189 пикселей Глубина цвета: 24</p>	 <p>Размеры: 115x264 пикселей Глубина цвета: 24</p>	 <p>Размеры: 275x401 пикселей Глубина цвета: 24</p>	

Эксперименты проводились на персональной ЭВМ со следующими характеристиками: центральный процессор Intel Pentium Duo T2390 с частотой 1,87 ГГц, объем оперативной памяти 1 Гб, объем видеопамати 256 Мб (ЭВМ с такими характеристиками может устанавливаться на мобильные робототехнические платформы [8]).

Получаемые с камеры изображения (кадры) могут иметь любой размер, выраженный в пикселях. Кадр является исходным изображением для распознавания объектов. Пример дескрипторов изображения показан на рис. 7. После этого в получаемом видеопотоке с камеры робота можно распознавать как подвижные (рис. 8), так и неподвижные (рис. 9) объекты, присутствующие в коллекции образов.



Рис. 7 Дескрипторы, полученные с помощью алгоритма «SURF»

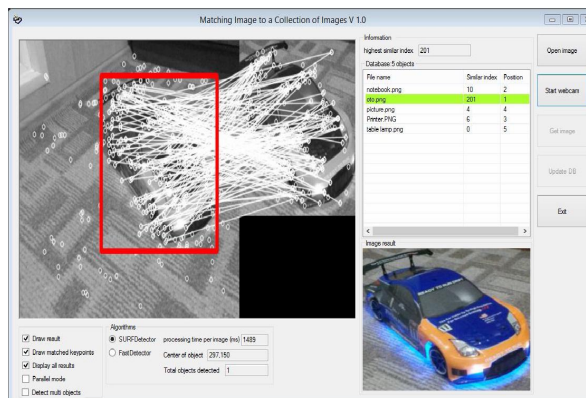


Рис. 8. Обнаружение подвижного объекта (радиоуправляемая машина)

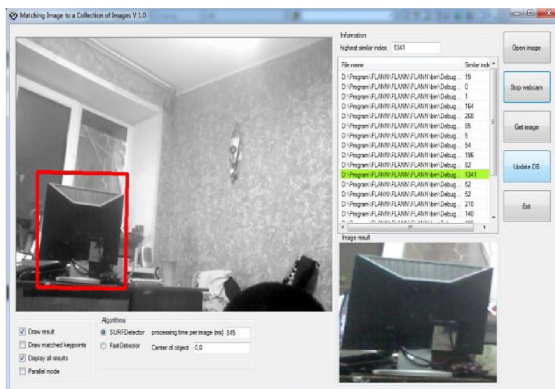


Рис. 9. Обнаружение неподвижного объекта (монитор)

Качество обнаружения неподвижного объекта значительно лучше, чем движущегося, так как в процессе перемещения изображение такого объекта, полученное камерой, может оказаться нечетким. Эксперименты проводились с различными коллекциями образов для неподвижных распознаваемых объектов. Предварительной обработки изображений (например, с применением средств увеличения четкости изображения или приведения контуров к хорошо распознаваемым геометрическим фигурам) не проводилось. Полученные результаты наглядно показаны на рис. 10 и в числовой форме сведены в табл. 2.

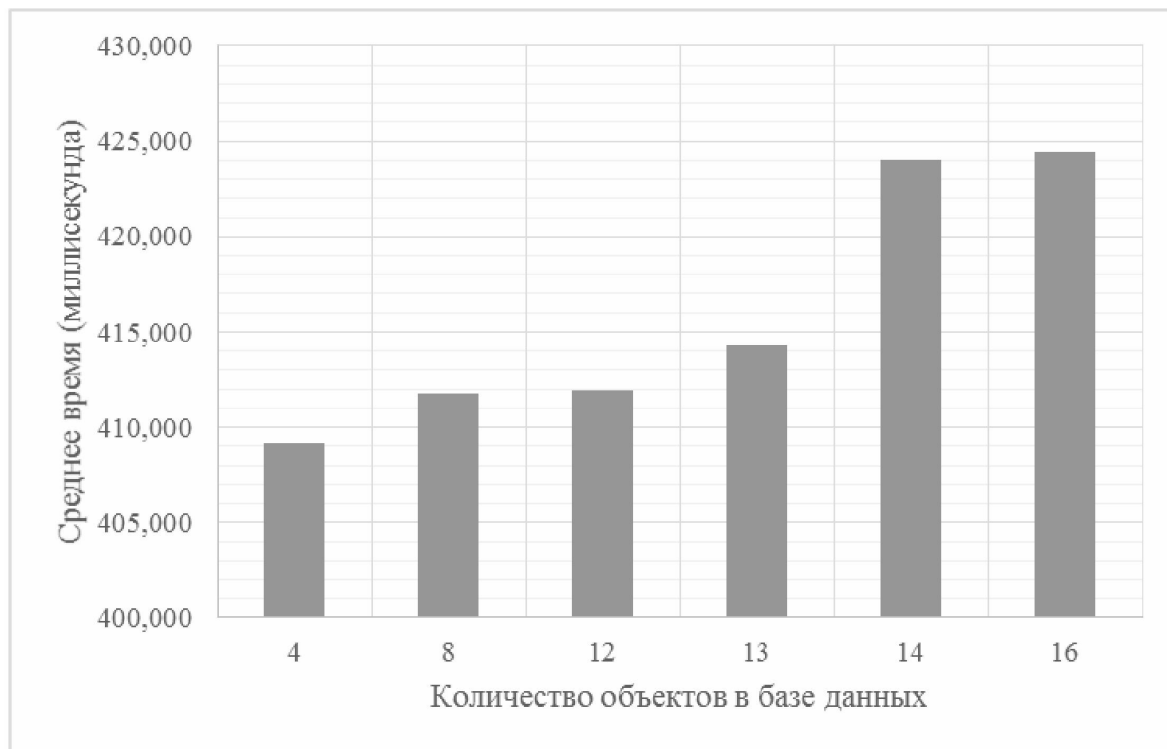


Рис. 10. Время обнаружения объекта с применением различных по размерам коллекций изображений

Таблица 2

Результаты экспериментов					
№	Количество объектов коллекции	Количество входных изображений	Ошибка	Точность (%)	Среднее время на обнаружение объекта (миллисекунды)
1	4	52	3	94	409,1
2	8	52	4	92	411,9
3	12	52	6	88	414,3
4	13	52	6	88	411,8
5	14	52	7	86	424,0
6	16	52	9	82	424,4
Среднее значение				88	415,9

Из таблицы 2 видно, что при увеличении количества объектов в 4 раза среднее время обнаружения объекта увеличивается только в 1,037 раза. При этом средняя точность обнаружения равна 88 % (под ошибкой в данном исследовании понимается то, что объект распознается неверно или не распознается, хотя его эталонное изображение есть в базе).

Таким образом, результаты вычислительных экспериментов модельного характера свидетельствуют о вычислительной эффективности (высокой скорости работы и малой доле ошибок распознавания) использования библиотеки «FLANN» и алгоритма «SURF» при их применении в составе программного обеспечения системы технического зрения мобильного робота (для обнаружения различных объектов в ЗиС, в том числе осуществляющих движение). Требуемые скорость и точность распознавания обеспечиваются и при движении самого робота. Кроме того, обнаружение объектов, присутствующих в коллекции изображений, позволяет роботу более эффективно ориентироваться в ЗиС с заранее не известной картой пространства.

Заключение. Научная новизна работы заключается в том, что впервые предложено совместное использование библиотеки FLANN и алгоритма SURF для распознавания объектов различных типов (как подвижных, так и неподвижных), обеспечивая тем самым требуемую точность и скорость ориентирования мобильных робототехнических платформ в ЗиС.

Библиотека FLANN может быть использована для нахождения соответствия изображения в коллекции образов. Эффективность нахождения объектов будет увеличиться в случае применения алгоритма SURF. Использование библиотеки FLANN совместно с алгоритмом SURF позволяет удовлетворить требованиям по скорости обнаружения объектов в режиме реального времени. Полученные в работе результаты позволяют рекомендовать использование библиотеки FLANN и алгоритма SURF в составе системы технического зрения робота и для решения задач управления его движением в реальном времени.

Список литературы

1. Бурдаков С. Ф. Системы управления движением колесных роботов / С. Ф. Бурдаков, И. В. Мирошник, Р. Э. Стельмаков. – Санкт-Петербург : Наука, 2001. – 227 с.
2. Голубкин И. А. Универсальная сенсорная подсистема мобильного колесного робота / И. А. Голубкин, И. А. Щербатов // Датчики и системы. – 2010. – № 8. – С. 32–35.
3. Голубкин И. А. Принципы разработки мультимедийных прикладных роботизированных устройств / И. А. Голубкин, С. А. Кирилин, И. А. Щербатов // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2010. – № 2. – С. 153–157.
4. Михайлова Е. Г. Приближенное индексирование многомерных объектов / Е. Г. Михайлова, Б. А. Новиков // Электронные библиотеки: перспективные методы и технологии, электронные коллекции : тр. XII Всерос. науч. конф. RCDL'2010 ; Казань, Россия 13–17 октября 2010 г. – Казань : Казанский университет, 2010. – С. 436–440.

5. Носков В. П. Навигация мобильных роботов по дальнометрическим изображениям / В. П. Носков, А. В. Носков // Мехатроника, автоматизация, управление. – 2005. – № 12. – С. 16–21.
6. Обзор информации о роботизированных комплексах. – Режим доступа: http://mchs.gov.by/_modules/_files/files/obzor_o_robotizirovannykh_kompleksah.pdf (дата обращения 11.11.2014), свободный. – Заглавие с экрана. – Яз. рус.
7. Обзор информации о FLANN. – Режим доступа: <http://www.cs.ubc.ca/research/flann/> (дата обращения 30.10.2014), свободный. – Заглавие с экрана. – Яз. рус.
8. Чернухин Ю. В. Нейросетевой подход к решению задачи локально навигации интеллектуальными мобильными роботами в условиях, приближенных к реальной среде / Ю. В. Чернухин, Ю. С. Доленко, П. А. Бутов // Известия Южного федерального университета. Технические науки. – 2013. – № 5 (142). – С. 80–84.
9. Bay H. Speeded-Up Robust Features (SURF) / H. Bay, A. Ess, T. Tuytelaars, Van Gool L. // Computer Vision and Image Understanding. – 2008. – June. – Vol. 110, iss. 3. – P. 346–359.
10. 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.
11. Bota S. Camera motion estimation using monocular and stereo-vision / S. Bota // Intelligent Computer Communication and Processing. – 2008. – P. 275–278.
12. Bradski G. Learning OpenCV / G. Bradski, A. Kaehler. – O'Reilly Media, 2008. – 576 p.
13. Forsyth D. A. Computer Vision: A Modern Approach / D. A. Forsyth, J. Ponce // Robotics and Autonomous Systems. – 2006. – Vol. 11, № 3. – P. 84–92.
14. Juan L. A Comparison of SIFT, PCA-SIFT and SURF/ L. Juan, O. Gwun // International Journal of Image Processing (IJIP). – 2009. – Vol. 3, № 4. – P. 143–152.
15. Kefer M. Evaluation of Kinect Depth Sensor for Use in Mobile Robotics / M. Kefer, W. Kubinger // Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium. Vienne, 2011. – Vol. 22, № 1.
16. k-d tree. – Available at: http://en.wikipedia.org/wiki/K-d_tree (accessed 30.10.2014).
17. Marius Muja. Fast approximate nearest neighbors with automatic algorithm configuration / Marius Muja, David G. Lowe // VISAPP International Conference on Computer Vision Theory and Applications. – 2009. – P. 331–340.
18. Ramisa A. Evaluation of the SIFT Object Recognition Method in Mobile Robots / A. Ramisa, S. Vasudevan, D. Aldavert, R. Toledo, Lopez de Mantaras R. – Available at: http://www.iiia.csic.es/~mantaras/sift_eval.pdf (accessed 30.10.2014).

References

1. Burdakov S. F., Miroshnik I. V., Stelmakov R. E. *Sistemy upravleniya dvizheniem kolesnykh robotov*. Saint Petersburg, Nauka, 2001. 227 p.
2. Golubkin I. A., Shcherbatov I. A. Universalnaya sensornaya podsystema mobilnogo kolesnogo robota [Universal touch subsystem of mobile wheeled robot]. *Datchiki i sistemy* [Sensors and Systems], 2010, no. 8, pp. 32–35.
3. Golubkin I. A., Kirilin S. A., Shcherbatov I. A. Printsipy razrabotki multimediynykh prikladnykh robotizirovannykh ustroystv [Principles of multimedia applications of robotic devices]. *Vestnik Astrakhan-skogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika i informatika* [Bulletin of Astrakhan State Technical University. Series: Management, Computer Science and Informatics], 2010, no. 2, pp. 153–157.
4. Mikhaylova Ye. G., Novikov B. A. Priblizhennoe indeksirovanie mnogomernykh obektov [Approximate indexing multi-dimensional objects]. *Elektronnye biblioteki: perspektivnye metody i tekhnologii, elektronnye kolleksii: trudy XII Vserossiyskoy nauchnoy konferentsii RCDL'2010* [Digital libraries: advanced methods and technologies, digital number of lectures: Proceedings of the XII All-Russian Scientific Conference RCDL'2010]. Kazan, Kazan University Publ., 2010, pp. 436–440.
5. Noskov V. P., Noskov A. V. Navigatsiya mobilnykh robotov po dalnometricheskimi izobrazheniyam [Navigation of mobile robots for dalnometricheskimi images]. *Mekhatronika, avtomatizatsiya, upravlenie* [Mechatronics, Automation, Control], 2005, no. 12, pp. 16–21.

6. Review of the information on robotic complexes. Available at: http://mchs.gov.by/_modules/_cfiles/files/obzor_o_robotizirovannih_kompleksah.pdf (accessed 11 November 2014). (In Russ.)
7. Review of information on FLANN. Available at: <http://www.cs.ubc.ca/research/flann/> (accessed 30 October 2014). (In Russ.)
8. Chernukhin Yu. V., Dolenko Yu. S., Butov P. A. Neyrosetevoy podkhod k reshetsiyu zadachi lokalno navigatsii intellektualnymi mobilnymi robotami v usloviyakh, priblizhennykh k realnoy srede [A neural network approach to the problem locally reshetsiyu mind-trivial navigation of mobile robots in conditions close to the real environment]. *Izvestiya Yuzhnogo federalnogo universiteta. Tekhnicheskie nauki [Southern Federal University. Technical Sciences]*, 2013, no. 5 (142), pp. 80–84.
9. Bay H., Ess A., Tuytelaars T., Van Gool L. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 2008, June, vol. 110, iss. 3, pp. 346–359.
10. Bonin-Font F., Ortiz A., Oliver G. Visual Navigation for Mobile Robots: a Survey. *Journal of Intelligent and Robotic System*, 2008, pp. 263–296.
11. Bota S. Camera motion estimation using monocular and stereo-vision. *Intelligent Computer Communication and Processing*, 2008, pp. 275–278.
12. Bradski G., Kaehler A. *Learning OpenCV*. O'Reilly Media, 2008. 576 p.
13. Forsyth D. A., Ponce J. Computer Vision: A Modern Approach. *Robotics and Autonomous Systems*, 2006, vol. 11, no. 3, pp. 84–92.
14. Juan L., Gwun O. A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 2009, vol. 3, no. 4, pp. 143–152.
15. 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*. Vienne, 2011, vol. 22, no. 1.
16. k-d tree. Available at: http://en.wikipedia.org/wiki/K-d_tree (accessed 30 October 2014).
17. Marius Muja, David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP International Conference on Computer Vision Theory and Applications*, 2009, pp. 331–340.
18. 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.iia.csic.es/~mantaras/sift_eval.pdf. (accessed 30 October 2014)

УДК 004.4

ПРОЕКТИРОВАНИЕ РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ В ГИБРИДНОЙ ОБЛАЧНОЙ ИНФРАСТРУКТУРЕ НА ОСНОВЕ ОБЪЕКТНО-РЕЛЯЦИОННОГО ОТОБРАЖЕНИЯ

Статья поступила редакцию 17.11.2014, в окончательном варианте 06.12.2014.

Лукьянчиков Олег Игоревич, преподаватель, Московский государственный университет приборостроения и информатики, 107996, Российская Федерация, г. Москва, ул. Стромынка, 20, e-mail: luk-it6@yandex.ru

Никольчев Евгений Витальевич, доктор технических наук, профессор, проректор по научной работе, Московский технологический институт, 119334, Российская Федерация, г. Москва, Ленинский пр-т, 38а; профессор Высшей школы системного инжиниринга, Московский физико-технический институт (государственный университет), 141700, Российская Федерация, Московская обл., г. Долгопрудный, Университетский пер., 9, e-mail: e_nikulchev@mti.edu.ru

Паяин Семен Владимирович, кандидат технических наук, Московский технологический институт, 119334, Российская Федерация, г. Москва, Ленинский пр-т, 38а, e-mail: sadsema@gmail.com

Плужник Евгений Владимирович, первый проректор, Московский технологический институт, 119334, Российская Федерация, г. Москва, Ленинский пр-т, 38а; аспирант, Московский государственный университет печати имени Ивана Федорова, 127550, Российская Федерация, г. Москва, ул. Прянишникова, 2А, e-mail: e_pluzhnik@mti.edu.ru