

Т. М. Волосатова, канд. техн. наук, доц., tamaravol@gmail.com, А. В. Козов, alexey.kozov@gmail.com, МГТУ им. Н. Э. Баумана

Особенности методов распознавания образов в автоматической системе управления поворотом мобильного робота

Рассмотрена задача построения системы компьютерного зрения для управления мобильным роботом. Разработаны основные требования к системе управления мобильным роботом, определен выбор операционной системы для мобильного робота. Приведен обзор и выполнено сравнение методов распознавания заданного объекта на изображении с использованием современных программных инструментов и с учетом ограничений мобильных вычислительных платформ. По итогам сравнения для реализации в прототипе системы управления был выбран метод контурной сегментации и последующего сравнения контуров. Реализация прототипа системы управления была выполнена в виде исполняемого приложения в среде ROS. Реализованная система успешно протестирована в робототехническом симуляторе Gazebo. Исследования возможности оценки расстояния до распознаваемого объекта, а также эффективность комбинаций различных методов могут быть дальнейшими направлениями развития данной работы.

Ключевые слова: компьютерное зрение, распознавание образов, мобильный робот, система управления, эффективность распознавания

Введение

Системы технического (или компьютерного) зрения являются неотъемлемой частью систем управления многих мобильных робототехнических комплексов [1]. С помощью методов компьютерного зрения может быть решен широкий спектр задач, связанных с ориентацией во внешнем пространстве, определением расстояний до объектов, визуальной одометрией, распознаванием объектов окружения. К таким задачам относится управление мобильным роботом на основе распознавания изображений, получаемых с бортовой камеры. На кадрах необходимо распознать и локализовать некоторый заранее заданный объект, после чего нужно управлять роботом таким образом, чтобы сохранять распознанный объект в поле зрения камеры робота, т.е. следить за объектом. В рассматриваемом устройстве изменяется только угловая скорость робота, а камера неподвижно закреплена на его корпусе, оптическая ось камеры соосна продольной оси робота. Также принято допущение о совпадении вершины угла обзора камеры γ с осью поворота робота. Угол α , на который робот должен повернуться, соответствует углу между направлением на объект и продольной осью робота (рис. 1).

В рассматриваемой задаче принято допущение, что распознаваемый объект обладает маркером — определенной графической меткой, образ которой априорно известен, по которой выполняется поиск и распознавание объекта. Для распознавания маркера на изображении существует большое число методов. Наиболее просты методы поиска по шаблону, которые отыскивают на изображении наиболее похожий на искомый образ фрагмент. Однако такие методы не обладают инвариантностью к искажениям распознавае-

мого образа. Методы на основе сегментации используют для распознавания не искомого образа целиком, а только его некоторых наиболее характерных частей. В последнее время получили широкое распространение методы категориального распознавания. Они обладают высокой эффективностью, но их практическое применение ограничено необходимостью предварительного обучения и тонкой настройки их параметров.

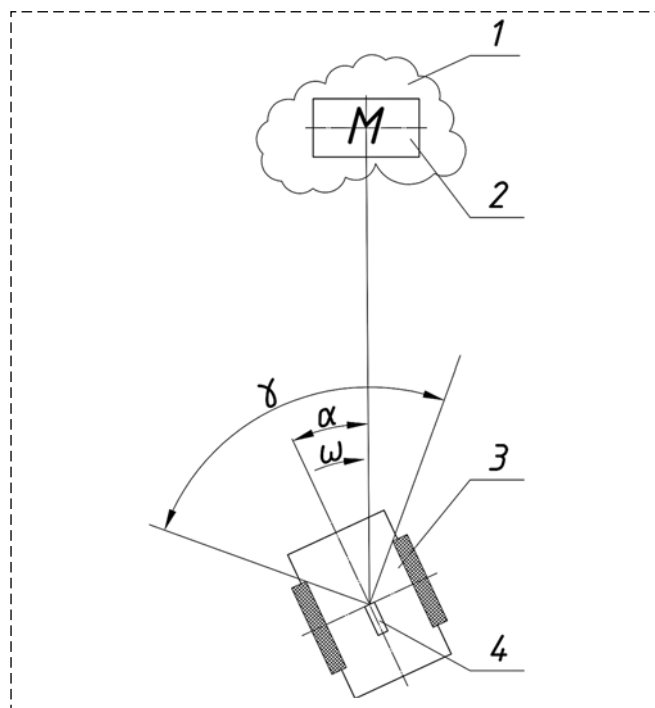


Рис. 1. Задача управления угловой скоростью мобильного робота:

1 — распознаваемый объект; 2 — маркер на объекте; 3 — мобильный робот; 4 — камера робота

Благодаря существенному прогрессу в области технического зрения сегодня имеется широкий спектр инструментов и методов для решения задач автоматизированного распознавания образов [2], а развитие микропроцессорных технологий позволяет оснащать даже сравнительно небольшие мобильные робототехнические комплексы мощными процессорами уровня Intel Core i7. Однако увеличение вычислительной мощности бортового процессора негативно сказывается на автономности комплекса из-за увеличения его энергопотребления. Поэтому на мобильных роботах предпочтительнее установка мобильных вычислительных систем на одном кристалле с высокой энергоэффективностью, но меньшей производительностью. Ограничения по вычислительным ресурсам можно обойти, если использовать облачные вычисления, когда наиболее ресурсоемкие задачи рассчитываются на удаленном сервере. При таком подходе необходимо решать дополнительные задачи, связанные с синхронизацией данных и обеспечением надежности канала связи с сервером. Операционной системой, в которой должна работать рассматриваемая система, является облегченная версия дистрибутива Ubuntu версии 14.04. Доступные вычислительные ресурсы робота соответствуют двоядерному процессору Intel Atom частотой до 1,5 МГц и одному гигабайту оперативной памяти.

Требование работы в реальном времени подразумевает, что обработка кадров с камеры робота будет проводиться быстрее, чем частота съемки кадров. Например, если камера работает с частотой 30 кадров в секунду, система управления должна успеть обработать отдельный кадр и сгенерировать управляющий сигнал за время, не превышающее 33 мс. При этом необходимо иметь резерв вычислительных ресурсов для решения других задач по управлению робототехническим комплексом. Выполнение таких ограничений возможно в том числе за счет изменения настроек камеры (уменьшения частоты или разрешения).

В настоящее время существует большое число различных программных инструментов, предназначенных для облегчения разработки программ для робототехнических систем. Некоторые инструменты (например, Carnegie Mellon Robot Navigation Toolkit Carmen) предназначены для решения специализированных задач, другие являются расширением операционной системы, обеспечивая взаимодействие между разными компонентами (URBI, The OrocOS Toolchain). Также существуют программные робототехнические системы, созданные специально для разработки на конкретном языке программирования. Следует отметить, что наибольшее распространение получили программные системы, расширяющие возможности операционной системы и предоставляющие многофункциональные интерфейсы взаимодействия для программных и аппаратных компонентов. Эти системы постепенно вытесня-

ют или включают в себя специализированные инструменты. К таким метаоперационным системам относится Robot Operating System (ROS) [3], которая на сегодняшний день является, де-факто, стандартом робототехники.

ROS — программная среда с открытым исходным кодом (лицензия BSD), которая предоставляет набор инструментов для разработки и большое число готовых модулей, необходимых для управления роботом. Распространенность и простота системы определяют выбор ROS в качестве единой среды взаимодействия между различными компонентами системы управления. Таким образом, система должна поддерживать совместимость с ROS.

Немаловажным требованием к системе управления роботом является то, что она должна функционировать без непосредственного участия человека-оператора. Его участие допускается в момент первоначальной настройки робота для выбора распознаваемого объекта из некоторого заранее заданного перечня. В дальнейшем система управления должна работать автономно.

1. Методы распознавания объектов

Основой для создания подсистем компьютерного зрения является наиболее популярная на сегодняшний день библиотека OpenCV (Open Source Computer Vision) [4]. OpenCV — кроссплатформенная библиотека алгоритмов технического зрения, обработки изображений и машинного обучения с открытыми исходными кодами. Среди библиотек компьютерного зрения (OpenVX, VIGRA, CCV, SimpleCV и др.) она поддерживает наибольшее число языков программирования и наибольшее число реализованных методов обработки изображений. Также она обладает полным набором базовых примитивов для реализации новых алгоритмов. Описанные далее методы распознавания рассмотрены в контексте применения классов и функций из библиотеки OpenCV версии 2.4.13.2.

Для идентификации распознаваемого объекта в решаемой задаче применяется маркер — некоторое заранее выбранное изображение с определенными формой, размером и цветами. Его можно рассматривать также как единицу специального искусственного графического языка. С помощью маркера оптическим путем может передаваться некоторый объем информации, зависящий от размера тезауруса. Кроме того, по маркеру может быть определено положение камеры в пространстве [5]. Для создания маркера могут быть использованы как существующие специальные графические языки (например, коды Data Matrix или QR), так и определенные буквенно-цифровые сочетания (примером могут служить системы государственных регистрационных автомобильных номерных знаков). В данной работе в качестве од-

Рис. 2. Пример использованных для распознавания маркеров

ного из возможных маркеров использовано изображение, приведенное на рис. 2.

Одним из простейших методов распознавания заданного образа на изображении является метод сравнения с шаблоном. Функцию поиска объекта по шаблону можно представить следующим образом: изображение шаблона последовательно накладывается на исходное (тестовое) изображение, и между ними попиксельно вычисляется корреляция, а результат вычислений записывается в некоторую матрицу. Эту матрицу можно считать матрицей соответствия шаблона тому или иному фрагменту тестового изображения. В полученной матрице можно найти максимальное (или, в зависимости от алгоритма вычисления коэффициента корреляции, минимальное) значение. Положение этого значения в матрице будет соответствовать расположению наиболее похожего на распознаваемый объект фрагмента тестового изображения. В библиотеке OpenCV присутствует функция `matchTemplate(...)`, которая реализует подобный механизм сравнения. Она поддерживает шесть различных алгоритмов расчета коэффициента корреляции [6]. Особенностью реализации функции `matchTemplate(...)` является зависимость результата ее выполнения от соответствия размеров шаблона и распознаваемого маркера на тестовом изображении. Для корректной работы требуется серия изменений масштаба шаблона в заданных пределах с некоторым шагом и последовательный поиск полученных образцов на тестовом изображении. В качестве результата распознавания может быть выбран фрагмент, имеющий наилучшую по всей серии корреляцию с шаблоном. При этом необходимо определить пороговое значение корреляции для предотвращения ложноположительных срабатываний. Если наилучшее значение ниже порогового, то это значит, что маркер на тестовом изображении не найден. Однако для алгоритмов расчета коэффициента корреляции без нормализации это пороговое значение будет зависеть от параметров изображения-маркера.

Использование сегментации представляет другой подход к распознаванию образов. Сегментация — процесс разделения цифрового изображения на несколько областей или секторов, при котором пиксели внутри одной области имеют некоторые общие визуальные характеристики [7]. Для распознавания образов может быть применена контурная сегментация изображения, т. е. разбиение по внешним границам объектов. Выделенные контуры можно сравнивать по каким-либо характеристикам с выделенными аналогичным способом контурами на маркере для определения

степени их сходства. В OpenCV для поиска контуров есть функция `findContours(...)`, работающая с бинарными изображениями. Функция имеет необязательный параметр `hierarchy` — вектор, содержащий информацию о топологии обрабатываемого изображения [8]. При работе в режиме построения иерархии для каждого обнаруженного контура будет доступна информация о контурах: "родитель" (внутри которого расположен текущий), первый "потомок" (расположен внутри текущего) и "сосед" (на одном уровне иерархии). Благодаря такой информации становится возможной быстрая фильтрация найденных контуров по соответствию иерархии контуров на маркере. Отобранные таким способом контуры можно сравнить с контурами маркера по некоторой характеристике, инвариантной к масштабированию и повороту, например по нормализованным центральным моментам [9]. В OpenCV реализована функция `matchShapes(...)`, которая вычисляет эти характеристики контуров, сопоставляет их и возвращает неотрицательное действительное число, обозначающее, насколько контуры соответствуют друг другу.

Сегментация изображения также может быть применена при выделении характерных признаков (ключевых точек) — некоторых характерных локальных особенностей. К простейшим признакам относятся углы и границы на изображении. Однако такие признаки чувствительны к масштабированию, поэтому обычно применяют методы детектирования признаков, использующие информацию об окрестности точки [10]. Далее необходимо получить описание (дескриптор) признака, позволяющее идентифицировать объект и найти соответствия между характерными точками инвариантно к преобразованиям изображения. Распознавание объекта в таком случае может быть основано на сопоставлении дескрипторов признаков маркера и обрабатываемого изображения с учетом преобразований масштаба и перспективы. OpenCV обладает полным набором инструментов для реализации описанного подхода: абстрактные классы для детектирования (`FeatureDetector`), получения дескрипторов (`DescriptorExtractor`), их сравнения (`DescriptorMatcher`) и функции определения возможных преобразований перспективы `findHomography(...)`. Присутствуют реализации таких методов, как SIFT (Scale-Invariant Feature Transform) и SURF (Speeded Up Robust Features) [11].

Решение задачи распознавания образов может быть основано на классификации объектов на изображении (категориальное распознавание). В этом случае необходимо определить набор признаков, по которым будет проводиться классификация, выбрать метод классификации и выполнить процедуру обучения классификатора. В качестве признаков могут выступать гистограммы направленных градиентов или примитивы Хаара. Для классификации можно воспользоваться, например, методом Виолы—Джонса, который

имеет высокую скорость и эффективность работы. Основу метода Виолы—Джонса составляют интегральное представление изображения по признакам Хаара, построение классификатора на основе алгоритма адаптивного бустинга и комбинирование классификаторов в каскадную структуру [12]. Представленный в OpenCV класс CascadeClassifier реализует метод Виолы—Джонса [13]. Он работает с предварительно обученным классификатором. Для обучения классификатора имеется утилита `opencv_traincascade`. На основе выборки из положительных (содержащих распознаваемый объект) и отрицательных (без распознаваемого объекта) примеров она формирует специальный XML-файл, описывающий обученный классификатор.

Для распознавания объектов также можно применять статистический анализ изображения. Например, для обнаружения объекта с вертикальными или горизонтальными контурами можно использовать детектор границ, и при проецировании на соответствующие оси по гистограммам яркости локализовать искомый объект. Однако для достижения инвариантности к поворотам и перспективным искажениям распознаваемого объекта необходимы дополнительные преобразования и проверки. Это снижает производительность и налагает ограничения на условия использования метода.

Также одним из вариантов построения системы распознавания, основанных на классификации, является использование искусственных нейронных сетей. Нейросетевой классификатор обладает высокой скоростью работы и робастностью, а главным недостатком является необходимость предварительного длительного обучения. Кроме того, необходима дополнительная работа по реализации и подбору оптимальных параметров нейронной сети. Указанные особенности затрудняют применение этого метода.

Сравнительный анализ эффективности методов распознавания

Для сравнения эффективности различных методов распознавания и локализации маркера разработано несколько специальных утилит [14]. Методы сравнивали на серии из 120 тестовых изображений (тестов) размером 1024 на 768 пикселей. Половина изображений содержала распознаваемый объект с маркером, остальные — нет. Тестовые изображения генерировались утилитой на основании двух отдельных изображений — переднего плана и фона. Передний план представляет собой снимок трехмерной сцены, построенной в робототехническом симуляторе Gazebo. На 3D-сцене могут быть представлены условные модели роботов, один из которых необходимо распознать по маркеру на борту, и несколько моделей окружения — дорога, здания. На таком изображении в графическом редакторе

однотонный фон был заменен на прозрачный. В качестве фоновых изображений были использованы фотографии реальной местности. Для получения тестового изображения образцы изображений переднего плана и фона совмещаются программой-генератором в различных комбинациях. Примеры полученных таким способом тестовых изображений представлены на рис. 3 (см. третью сторону обложки).

Утилита, выполняющая сравнение методов распознавания, в качестве параметра при запуске принимает путь к изображению-маркеру и список тестовых изображений. Во время работы она поочередно загружает и обрабатывает тесты согласно списку. По заданному алгоритму утилита определяет положение маркера на тестовом изображении и сравнивает координаты предполагаемого положения ("—1; —1" для случая, если маркер не обнаружен) с координатами реального положения, известного для каждого теста. При этом для каждого изображения фиксируется время работы алгоритма распознавания, а в конце выводится среднее по всей серии тестов время в миллисекундах и процент ошибок распознавания для каждого алгоритма.

Как отмечалось ранее, используемая в OpenCV функция сравнения с шаблоном использует один из шести реализованных алгоритмов расчета коэффициента корреляции. По результатам испытаний хорошие результаты показал алгоритм нахождения нормализованного коэффициента кросс-корреляции по формуле

$$C(i, j) = \frac{\sum_{i', j'} T(i', j') A(i + i', j + j')}{\sqrt{\sum_{i', j'} T(i', j')^2 \sum_{i', j'} A(i + i', j + j')^2}},$$

где $T(i, j)$ — (i, j) -й элемент матрицы дискретного изображения-образца; $A(i, j)$ — (i, j) -й элемент матрицы дискретного тестового изображения. Наилучшие результаты достигаются при 9...11 шагах изменения масштаба шаблона, дальнейшее увеличение числа шагов не приводит к увеличению точности (рис. 4). Хотя сама функция поиска по шаблону выполняется достаточно быстро (в зависимости от алгоритма, 40...50 мс), необходимость дополнительных изменений масштаба и повторного поиска существенно снижает общее быстродействие метода. При используемых размерах тестовых изображений применение этого метода затруднено из-за его низкой производительности.

Применение контурной сегментации требует предобработки изображения: размытия для уменьшения числа контуров и бинаризации с выделением границ. Для размытия была использована доступная в OpenCV функция `blur(...)`, для бинаризации — функция преобразования цвета `cvtColor(...)` и детектор границ `Canny(...)`. Оптимальное значение радиуса размытия тестового изображения составляет 3...4 пикселя, увеличение этого значения существенно увеличивает процент

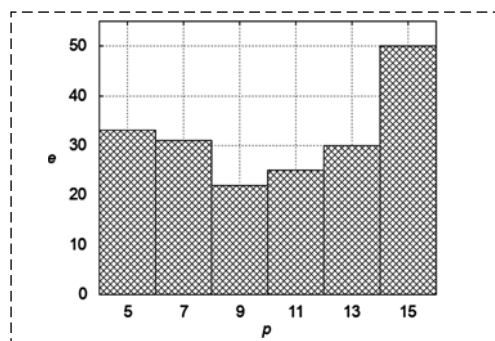


Рис. 4. Зависимость числа ошибок распознавания маркера от числа шагов изменения масштаба шаблона при использовании алгоритма нормализованного коэффициента кросс-корреляции:

p — число шагов изменения масштаба; e — процент ошибок распознавания

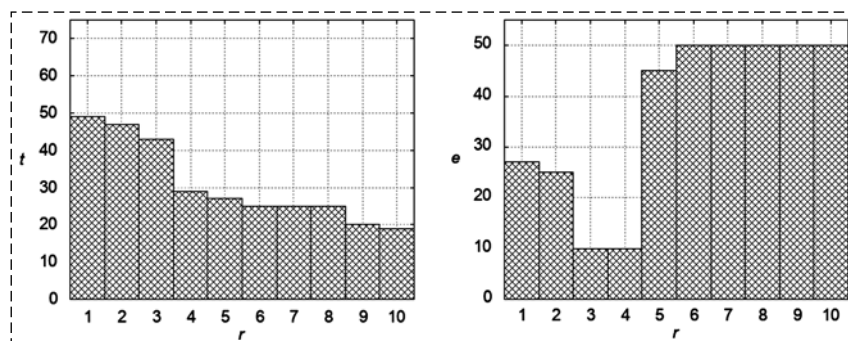


Рис. 5. Влияние величины пятна рассеяния (радиуса размыва тестового изображения) на скорость и качество распознавания:

r — радиус пятна рассеяния в пикселях; t — время обработки в миллисекундах; e — процент ошибок распознавания

ошибок (рис. 5). При сравнении контуров малого размера функцией `matchShapes(...)` могут быть получены высокие значения сходства. По этой причине контуры малого размера (до 15 пикселей) исключаются из числа сравниваемых для предотвращения ложноположительных срабатываний метода. Главным недостатком метода является его слабая устойчивость к помехам: зашумленность или частичная видимость маркера на тестовом изображении могут привести к невозможности корректного распознавания. Достоинствами же являются высокое быстродействие и хорошая точность распознавания в благоприятных условиях.

Эксперименты с различными реализациями представленных в OpenCV методов поиска объекта по характерным признакам показали их низкое быстродействие и невысокую эффективность, так как используемые маркеры не обладают достаточным числом характерных точек, на основании которых можно однозначно определить наличие или отсутствие маркера на тестовом изображении.

Сравнение различных методов распознавания

Метод	Достоинства	Недостатки
Сравнение с шаблоном	Простая реализация	Низкая скорость обработки, частые ложные срабатывания
Сравнение контуров	Простая реализация, высокая скорость работы, хорошая надежность	Метод может не работать при частичной видимости маркера на изображении
Выделение характерных признаков	—	Низкая скорость работы, не работает при используемых маркерах
Применение каскадного классификатора	Высокая скорость работы, наилучшая надежность	Длительное время обучения, необходимость дополнительной работы по подготовке обучающей выборки для каждого маркера

Построенная с помощью каскадного классификатора система распознавания обладает достаточным быстродействием (до 150 мс для обработки одного тестового изображения) и высокой надежностью, в том числе для случаев зашумленного изображения или частичного перекрытия маркера другим объектом. Главным недостатком такого подхода является необходимость предварительного обучения, для которого нужно большое число положительных и отрицательных примеров. Само обучение может занимать в зависимости от сложности маркера от нескольких часов до суток при обработке на персональном компьютере среднего уровня, что может существенно осложнить подбор оптимального маркера.

Достоинства и недостатки рассмотренных методов приведены в таблице. По итогам сравнения для реализации в прототипе системы управления был выбран метод контурной сегментации и последующего сравнения контуров. Также можно отметить, что хорошим потенциалом могут обладать комбинации методов, например, предварительная локализация возможного местоположения маркера с помощью контурной сегментации и последующее сравнение с шаблоном, масштаб которого изменен в соответствии с размерами контура.

2. Реализация системы управления роботом

Система управления поворотом робота должна работать в среде ROS, что налагает определенные требования к ее структуре. Основу любого ROS-приложения составляют пакеты, объединяющие исполняемые файлы, библиотеки, файлы конфигурации и другие ресурсы. Основной функциональной единицей системы является узел (node), работающие узлы можно представить в виде графа связей одноранговых точек-процессов, связанных через инфраструктуру ROS. Такие узлы могут обмениваться информацией посредством тем (topic), сервисов (service) и параметров (parameter). Тема представляет собой асинхронный поток данных, который генерируется од-

ним или несколькими издателями (publisher) и на который могут быть подписаны узлы-читатели (subscriber). Сервисы реализуют синхронный обмен информацией по схеме "запрос—ответ", а параметры позволяют сохранять данные в некотором общем хранилище — сервере параметров.

Реализация прототипа системы управления была выполнена в виде одного ROS-пакета. Этот пакет содержит исходный код программы исполняемого модуля, файлы запуска и конфигурации. Исполняемый файл (узел) написан на C++ с использованием средств ROS.

При запуске реализованного прототипа программа выполняет загрузку изображения заранее подготовленного маркера-образца в формате OpenCV. Предобработка включает перевод изображения в оттенки серого и размытие с помощью функции OpenCV `blur(...)`. Далее происходит обнаружение и расчет характеристик контуров. Изображение с камеры робота подвергается предобработке аналогично маркеру-образцу. На подготовленном таким образом кадре происходит обнаружение контуров, по параметрам которых выполняется распознавание. Для уменьшения числа сравнений активно используется иерархия: для сравнения отбираются только те контуры, которые проходят простые проверки на наличие одного общего контура-родителя. Проверка формы контура-родителя также позволяет отсеять большое число неперспективных контуров (например, слишком вытянутых в ширину или в высоту). Еще одним реализованным фильтром является проверка на минимальный размер — если контур меньше 15 пикселей по меньшему измерению, велика вероятность ложного срабатывания при сравнении моментов, поэтому такие контуры исключаются. Среди обнаруженных описанным способом серий одноранговых по топологии контуров выбирается та группа, которая имеет наибольшее среднее значение соответствия контурам маркера-образца. Если такое значение меньше 0,5, то считается, что распознаваемый образ не найден на текущем кадре.

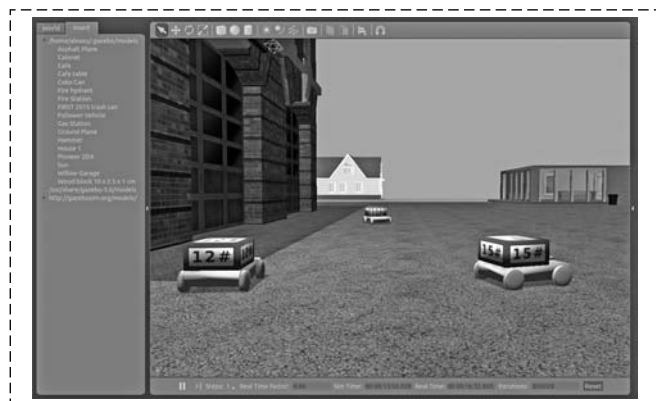


Рис. 6. Испытание реализованной системы управления в робототехническом симуляторе Gazebo. Представлена трехмерная сцена с моделями трех роботов, модель по центру разворачивается на месте в соответствии с перемещениями модели справа

После локализации маркера на кадре можно определить смещение центра распознанного маркера относительно центра кадра. В простейшем случае, если известна информация об угле обзора камеры по горизонтали, то, пренебрегая нелинейными искажениями оптического объектива, можно рассчитать точное направление на объект с маркером. После этого можно получить значение угловой скорости ω , которую необходимо сообщить роботу для поворота за заданное время (с учетом ограничений допустимой скорости). В случаях, когда информация об углах обзора камеры недоступна, необходимые значения угловой скорости можно подобрать экспериментально, наибольшие значения назначаются при максимальном смещении маркера от центра кадра по горизонтальной оси, а при совпадении их центров принимаются равными нулю. В случаях, когда распознаваемый объект движется относительно робота, необходимы вычисления поправок значений угловой скорости для компенсации возможного запаздывания поворота [15]. Реализованная система успешно протестирована в робототехническом симуляторе Gazebo (рис. 6).

Заключение

Для решения задачи по созданию автоматической системы управления мобильным роботом на основе компьютерного зрения было проведено сравнение различных методов распознавания заданного объекта на изображении. Получены показатели эффективности и производительности для четырех различных методов: сравнение с шаблоном, сравнение контуров, сопоставление ключевых точек, метод Виолы—Джонса. Метод, основанный на контурной сегментации и сравнении контуров, обладает оптимальным сочетанием достоинств и недостатков для применения в прототипе системы управления поворотом мобильного робота. При испытаниях реализованного прототипа с использованием робототехнического симулятора этот метод показал хорошую эффективность и надежность. Исследования возможности оценки расстояния до распознаваемого объекта, а также эффективность комбинаций различных методов могут быть дальнейшими направлениями развития данной работы.

Список литературы

1. Буняков В. А., Юревич Е. И. Техническое зрение в робототехнике. СПб.: Астерион, 2008. 67 с.
2. Визильтер Ю. В., Желтов С. Ю., Бондаренко А. В., Осоков М. В., Моржин А. В. Обработка и анализ изображений в задачах машинного зрения. М.: Физматкнига, 2010. 672 с.
3. Mahtani A., Sanchez L., Fernandez E. Effective Robotics Programming with ROS. — Packt Publishing, 2016. 468 с.
4. Bradski G., Kaehler A. Learning OpenCV: Computer vision with the OpenCV library. — O'Reilly Media, Inc., 2008.
5. Гудаев О. А. Распознавание маркеров расширенной реальности ARGET робототехнической системой // Известия Южного федерального университета. Технические науки. 2006. Т. 71. № 16.

6. **OpenCV 2.4.13.1** documentation — Object Detection, электронный ресурс http://docs.opencv.org/2.4/modules/imgproc/doc/object_detection.html (дата обращения 31.01.2017).
7. **Arbelaez P., Maire M., Fowlkes C., Malik J.** Contour detection and hierarchical image segmentation // *IEEE transactions on pattern analysis and machine intelligence*. 2011. Т. 33. N 5. С. 898—916.
8. **OpenCV 2.4.13.1** documentation — Structural Analysis and Shape Descriptors, электронный ресурс http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html (дата обращения 31.01.2017).
9. **Hu M. K.** Visual pattern recognition by moment invariants // *IRE transactions on information theory*. 1962. Т. 8. N. 2. С. 179—187.
10. **Lowe D. G.** Object recognition from local scale-invariant features // *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. — IEEE, 1999. Т. 2. С. 1150—1157.
11. **OpenCV 2.4.13.1** documentation — features2d. 2D Features Framework, электронный ресурс <http://docs.opencv.org/2.4/modules/features2d/doc/features2d.html> (дата обращения 31.01.2017).
12. **Viola P., Jones M.** Rapid object detection using a boosted cascade of simple features // *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. IEEE, 2001. Т. 1. С. 1-511—1-518.
13. **OpenCV 2.4.13.1** documentation — Cascade Classification, электронный ресурс http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html (дата обращения 31.01.2017).
14. **Comparison of object detection methods**, электронный ресурс <http://github.com/vvozokk/object-detecting> (дата обращения 31.01.2017).
15. **Benavidez P., Jamshidi M.** Mobile robot navigation and target tracking system // *System of Systems Engineering (SoSE), 2011 6th International Conference on*. IEEE, 2011. С. 299—304.

Features of Pattern Recognition Methods for the Turn Control System of Mobile Robot

T. M. Volosatova, tamaravol@gmail.com✉, **A. V. Kozov**, alexey.kozov@gmail.com,
Bauman Moscow State Technical University, Moscow 105005, Russian Federation

*Corresponding author: Volosatova T. M., Ph. D., Associate Professor,
Bauman Moscow State Technical University, Moscow, 105005, Russian Federation,
e-mail: tamaravol@gmail.com*

The paper considers the problem of designing the automatic control system of robot based on computer vision. The paper contains the problem formulation of designing the computer vision system to control mobile robot and describes restrictions and basic requirements for control system. The choice of the operating system for mobile robot is described. The paper also contains overview and comparison of recognition methods of the specified object in the image using modern program tools and considering restrictions of mobile computing platforms. The following methods have been analyzed: detection using a marker, pattern matching, image detection and contour matching, feature detections, object classification, method of statistical image analysis, application of artificial neural networks. The paper describes possible ways to implement these methods using OpenCV library. Experiments with OpenCV implementations have been made, and also main advantages and disadvantages of compared methods have been determined. As the result of comparison the method of image segmentation and contour matching has been chosen to be implemented in control system prototype. The paper also describes the implemented prototype of the mobile robot angular velocity control system, which is based on pattern recognition system. The implemented system has been tested in a robotic simulator Gazebo. Research of possibility of distance measurement to a recognized object, as well as research of the effectiveness of different methods combinations can be further directions of development of this work.

Keywords: computer vision, pattern recognition, mobile robot, control system, recognition efficiency

For citation:

Volosatova T. M., Kozov A. V. Features of Pattern Recognition Methods for the Turn Control System of Mobile Robot, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2018, vol. 19, no. 2, pp. 104—110.

DOI: 10.17587/mau.19.104-110

References

1. **Bunyakov V. A., Yurevich E. I.** *Tekhnicheskoe zrenie v robototekhnike* (Machine vision in robotics), St. Petersburg, Asterion, 2008, 67 p. (in Russian).
2. **Vizilter Yu., Zheltov S. Yu., Bondarenko A. V., Ososkov M. V., Morgins A. V.** *Obrabotka i analiz izobrazhenij v zadachah mashinnogo zreniya* (Processing and analysis of images in tasks of machine vision), Moscow, Fizmatkniga, 2010, 672 p. (in Russian).
3. **Mahtani A., Sanchez L., Fernandez E.** *Effective Robotics Programming with ROS*, Packt Publishing, 2016. 468 c.
4. **Bradski G., Kaehler A.** *Learning OpenCV: Computer vision with the OpenCV library*, O'Reilly Media, Inc., 2008.
5. **Gudaev O. A.** *Raspoznavanie markerov rasshirennoj real'nosti ARGET robototekhnicheskoy sistemoj* (Recognition of augmented reality markers ARGET with robotic system), *Izvestiya Yuzhnogo federal'nogo universiteta. Tekhnicheskie nauki*, 2006, vol. 71, no. 16 (in Russian).
6. **OpenCV 2.4.13.1** documentation — Object Detection, available at: http://docs.opencv.org/2.4/modules/imgproc/doc/object_detection.html
7. **Arbelaez P., Maire M., Fowlkes C., Malik J.** Contour detection and hierarchical image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, vol. 33, no. 5, pp. 898—916.
8. **OpenCV 2.4.13.1** documentation — Structural Analysis and Shape Descriptors, available at: http://docs.opencv.org/2.4/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html
9. **Hu M. K.** Visual pattern recognition by moment invariants, *IRE Transactions on Information Theory*, 1962, vol. 8, no. 2, pp. 179—187.
10. **Lowe D. G.** Object recognition from local scale-invariant features, *Computer vision, The proceedings of the seventh IEEE international conference on*, 1999, vol. 2, pp. 1150—1157.
11. **OpenCV 2.4.13.1** documentation — features2d. 2D Features Framework, available at: <http://docs.opencv.org/2.4/modules/features2d/doc/features2d.html>
12. **Viola P., Jones M.** Rapid object detection using a boosted cascade of simple features, *Computer Vision and Pattern Recognition, Proceedings of the 2001 IEEE Computer Society Conference on*, 2001, vol. 1. pp. 1-511—1-518.
13. **OpenCV 2.4.13.1** documentation — Cascade Classification, available at: http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html
14. **Comparison of object detection methods**, available at: <http://github.com/vvozokk/object-detecting>
15. **Benavidez P., Jamshidi M.** Mobile robot navigation and target tracking system, *System of Systems Engineering, 2011 6th International Conference on*. IEEE, 2011, pp. 299—304.