

МЕХАНИКА

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

УДК 004.93

Алгоритмы точной локализации объектов по данным визуальных сенсоров промышленных роботов

А. Ю. Гордилов, А. В. Шабурова

Пермский государственный национальный исследовательский университет

Россия, 614990, г. Пермь, ул. Букирева, 15

aleksey.gorodilov@gmail.com | alexshaburova@gmail.com

Рассматривается проблема локализации объектов на изображении, возникающая при применении промышленных роботов. Предложен метод, позволяющий идентифицировать объект и определить его положение в пространстве вне зависимости от ракурса съемки. Разработанный алгоритм включает в себя преобразование Хафа для определения потенциальных вершин объекта, отсеивание ложных вершин на основе методов кластеризации и оптимизированный перебор вариантов сопоставления модели и изображения объекта. В результате создан прототип системы, позволяющий (на примере куба) определить углы поворота объекта в пространстве по его изображению.

Ключевые слова: локализация объекта; распознавание образов; преобразование Хафа; кластеризация; центральное проецирование; углы Эйлера.

DOI: 10.17072/1993-0550-2017-4-19-24

Введение

Задача повышения производительности труда является одной из важнейших задач производства, которая может быть выполнена, в том числе, за счет широкого применения промышленных роботов. Под промышленным роботом понимают автоматическую машину, состоящую из исполнительного устройства в виде манипулятора, имеющего несколько степеней подвижности, и перепрограммируемого устройства для выполнения в производственном процессе двигательных или управляющих функций [1].

Одной из основных задач, связанных с применением роботов, является задача локализации объектов, с которыми роботы должны взаимодействовать. Высокий уровень автономности роботов требует качественного

программного обеспечения, которое должно отличаться высокой точностью локализации, а также иметь возможность работать с моделями различных объектов, что значительно повысит уровень адаптивности таких устройств. Поэтому создание программного обеспечения, соответствующего вышеперечисленным требованиям, является актуальной задачей.

Взаимодействие промышленных роботов с объектами осуществляется при помощи различных сенсоров (как правило, визуальных). Данные, получаемые с визуальных сенсоров промышленных роботов, могут быть представлены по-разному в зависимости от того, с каким сенсором работает система. Зачастую таким сенсором является обычная камера, поэтому в качестве данных выступает изображение или серия изображений.

1. Задача локализации

В настоящее время известно немало систем машинного зрения, разрабатываемых несколькими компаниями [2, 3, 4, 5]. В основном такие системы работают с объектами конкретных (как правило, достаточно простых) форм и не могут быть оперативно (без переписывания исходного кода) перенастроены на объекты другой формы, что существенно ограничивает возможности их применения. Данная ситуация связана с тем, что используемые в системах алгоритмы локализации неустойчивы к аффинным преобразованиям, способным сильно изменить изображение объекта, а кроме того, используемые алгоритмы не допускают частичного перекрытия анализируемых объектов. Таким образом, в разрабатываемой системе объект должен задаваться трехмерной моделью, а изменение его описания не должно повлечь за собой изменений в алгоритме.

Для демонстрации работоспособности предлагаемых методов в рамках данного исследования в качестве объекта выбран обычный однотонный куб, а в качестве его трехмерной модели – каркасная модель. Таким образом, объект представляется описанием восьми вершин куба.

Как было отмечено во введении, на вход системе поступает изображение. Локализация объекта на этом изображении подразумевает обнаружение объекта, определение координат его расположения относительно какой-либо заданной неподвижной точки (например, самой камеры), а также определение углов поворота объекта относительно выбранных осей.

Таким образом, для решения задачи локализации объекта на изображении необходимо выполнить следующие шаги:

1. Распознать границы объекта. В случае куба границами являются ребра этого куба.
2. Определить координаты объекта, основных точек модели этого объекта. В случае куба таковыми точками являются вершины куба.
3. Определить углы Эйлера поворота объекта в пространстве.

При этом к алгоритму локализации предъявляются следующие требования:

- 1) высокая вероятность успешной локализации объекта;

- 2) высокая скорость работы;
- 3) возможность работать с изображениями любого качества.

2. Распознавание ребер куба

Ребра куба являются границей между кубом и фоном либо границей между плоскостями куба с разной освещенностью, поэтому для их распознавания можно воспользоваться алгоритмами выделения границ. Для распознавания границ на изображении существуют следующие алгоритмы:

- 1) сопоставление шаблонов (pattern matching);
- 2) метод скользящего окна;
- 3) преобразование Хафа;
- 4) сумка визуальных слов.

Отметим, что метод, основанный на сопоставлении шаблонов, является весьма медленным и не учитывает возможность изменения ракурса объекта. Метод скользящего окна также достаточно медленный, требует подбора шагов перемещения, не обеспечивает необходимой точности. Алгоритмы на основе ключевых точек не гарантируют требуемой вероятности успешного распознавания ввиду особенностей рассматриваемого объекта (однотонный куб, как и многие реальные промышленные детали, не имеет никаких ярких признаков, поэтому уникальные ключевые точки выделить затруднительно).

Таким образом, для решения поставленной задачи был выбран алгоритм Хафа [6], поскольку он наиболее часто применяется для распознавания параметрически задаваемых объектов, устойчив к изменению ракурса и имеет высокую вероятность успешного распознавания.

3. Определение вершин куба

Поскольку любая вершина куба является концом нескольких ребер, чтобы определить на изображении все вершины куба, предлагается найти прямые, на которых лежат его ребра, а после определить точки их пересечения.

Реализованный в библиотеке OpenCV алгоритм преобразования Хафа возвращает набор векторов (x_1, y_1, x_2, y_2) , где (x_1, y_1) и (x_2, y_2) являются конечными точками каждого обнаруженного сегмента прямой. В идеальном случае эти координаты являются концами ребер, т. е. как раз вершинами. Однако в силу низкого качества изображения и частичного

перекрытия объекта эти значения могут существенно отличаться от реальных координат вершин. В частности, граница может быть распознана не целиком, а по частям, тогда полученные координаты будут лишь координатами какой-то внутренней точки ребра.

Для улучшения точности предлагается по полученным двум точкам восстановить прямую (на ней лежит ребро куба), а затем найти все точки пересечения прямых.

В связи с тем, что рассматриваются все возможные пересечения распознанных линий, то возникают лишние точки в результате пересечения прямых, задающих несмежные грани куба (эти точки оказываются за пределами куба).

Чтобы отбросить такие точки пересечения, продолжим работу только с точками, находящимися в окрестности конечных точек каждого обнаруженного сегмента прямой.

Помимо этого, ввиду недостаточной четкости изображения существует вероятность того, что алгоритм Хафа на одном ребре вместо одной прямой распознает несколько прямых с малым углом между ними, как это показано на рис. 1. Это приводит к появлению лишних точек пересечения в области расположения вершин куба.

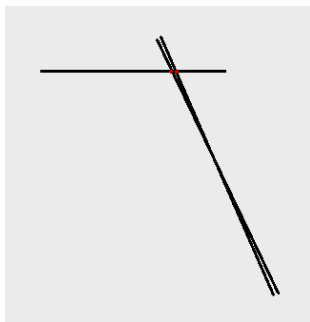


Рис. 1. Множественное пересечение

Решить данную проблему можно выделив в множестве полученных точек кластеры и заменив точки каждого кластера одной усредненной. Для этого воспользуемся простым алгоритмом выявления кластеров, описание которого приведено в [7].

Данный алгоритм является подходящим, так как не требует указания первоначального количества кластеров, а также достаточно точно определяет кластеры вокруг действительных вершин куба.

4. Определение углов поворота

Следующим этапом в решении задачи является определение углов поворота объекта на изображении, т. е. решение задачи, обратной к той, которая возникает при визуализации. При визуализации для поворота объекта на заданный угол и построении его проекции на экран используются известные матричные функции преобразования. Сложность обратной задачи заключается в том, что ее непосредственное решение влечет необходимость решения сложной системы уравнений, и известных формул для получения исходных координат вершин нет.

Исходя из сказанного, в данном исследовании предлагается следующий подход. Методом последовательных приближений будем перебирать различные углы поворота, выполнять рендеринг полученной модели объекта и сравнивать полученное изображение с анализируемым.

Решением задачи будет тот набор углов, при котором полученные в результате рендеринга координаты вершин наиболее точно совпадают с координатами вершин на анализируемом изображении.

Далее опишем, как будет выполняться поворот и проецирование модели.

Поворот вокруг произвольной оси в пространстве выполняется с помощью комбинации параллельного переноса и поворотов вокруг координатных осей. Основная идея заключается в том, чтобы совместить произвольную ось вращения с одной из координатных осей и выполнить поворот вокруг нее.

Предположим, что произвольная ось в пространстве проходит через точку (x_0, y_0, z_0) с направляющим вектором (c_x, c_y, c_z) . Поворот вокруг этой оси на некоторый угол δ выполняется по следующему алгоритму:

- 1) выполнить параллельный перенос так, чтобы точка (x_0, y_0, z_0) находилась в начале системы координат;
- 2) выполнить повороты вокруг осей x и y так, чтобы ось вращения совпала с осью z ;
- 3) выполнить поворот на угол δ вокруг оси z ;
- 4) выполнить преобразование, обратное шагу 2;
- 5) выполнить параллельный перенос обратный шагу 1.

Математически данная комбинация может быть записана следующим образом:

$$[X^*] = [X] [Tr] [R_x] [R_y] [R_z] [R_y]^{-1} [R_x]^{-1} [Tr]^{-1},$$

где:

- $[X^*]$ – преобразованное тело;
- $[X]$ – исходное тело;
- $[Tr]$ – матрица перемещения;
- $[R_x]$ – матрица поворота вокруг оси x ;
- $[R_y]$ – матрица поворота вокруг оси y ;
- $[R_z]$ – матрица поворота на произвольный угол (поворот вокруг оси z);

$[R_y]^{-1}$; $[R_x]^{-1}$; $[Tr]^{-1}$ – соответствующие обратные матрицы.

Для сопоставления модели с изображением необходимо определить двумерные координаты вершин, для этого воспользуемся центральным проецированием. За центр проекции примем точку, в которой находится камера, а за плоскость проекции – плоскость $Z = 0$. Таким образом, задача проецирования сводится к нахождению точки пересечения прямой и плоскости.

Чтобы оптимизировать перебор углов поворота объекта, воспользуемся методом последовательных приближений.

Для этого зададим изначальные углы поворота α , величины которых соответствуют половине длины отрезка рассматриваемого диапазона $[\theta_L; \theta_R]$. Зададим текущий шаг изменения углов σ , равный четверти длины отрезка диапазона. Далее запустим двойной цикл. Во внешнем цикле будем постепенно уменьшать шаг изменения углов σ , а во внутреннем – перебирать все возможные углы из диапазона $[\theta_L; \theta_R]$ с шагом σ .

На каждой внутренней итерации будем определять, насколько полученный результат рендеринга модели соответствует анализируемому изображению. Для этого каждой вершине, найденной на изображении, сопоставим ближайшую вершину модели, используя в качестве меры близости евклидово расстояние между точками. Если сумма этих расстояний будет меньше текущего минимума, то для следующей итерации внешнего цикла установим рассматриваемый диапазон углов в окрестности текущего: $[\theta_L; \theta_R] \leftarrow [\alpha - \sigma; \alpha + \sigma]$.

При завершении каждой внешней итерации будем уменьшать в два раза текущий шаг: $\sigma \leftarrow \sigma / 2$. Работа алгоритма продолжится до тех пор, пока текущий шаг σ не станет меньше заранее заданной точности ϵ .

5. Результаты

Описанный алгоритм реализован на языках C++ и C#, в реализации использована библиотека OpenCV. Для подготовки тестовых изображений была использована библиотека OpenGL.

Программа состоит из отдельных модулей. Модуль "Распознаватель" предназначен для распознавания объекта на изображении и его границ. Для этого использован алгоритм Хафа, реализованный в библиотеке OpenCV. В силу перспективных искажений прямые, содержащие каркас куба, пересекались не только в вершинах куба. Поэтому была реализована функция, отсеивающая ненужные точки: прежде всего, координаты которых находятся за пределами изображения, а после – координаты которых находятся за пределами найденных линий. Затем выполняется кластеризация, полученные координаты вершин сохраняются в текстовый файл (рис. 2).

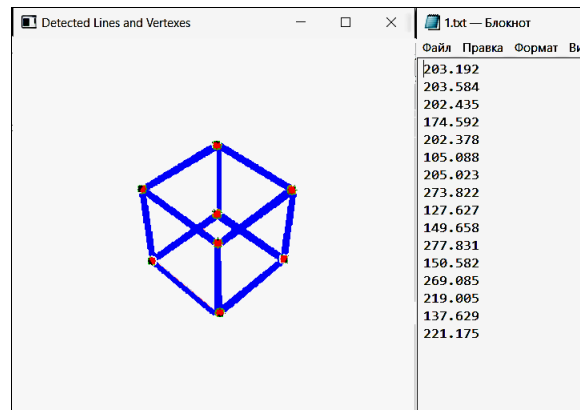


Рис. 2. Результат поиска вершин

Второй модуль, "Анализатор", предназначен для создания тестовых изображений и определения угла поворота объекта на тестовом изображении. Для визуальной оценки результата определения углов отрендеренная модель отображается на форме в элементе PictureBox рядом с исходным анализируемым изображением.

Таким образом, созданная программа позволяет создавать тестовые изображения с заранее известным положением и углами поворота куба (рис. 3: модель куба свободно вращается при помощи мыши), а также анализировать произвольное загружаемое изображение, в том числе, разумеется, подготовленное тестовое (рис. 4).

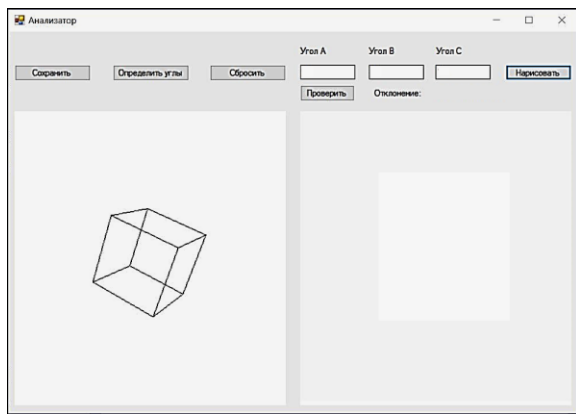


Рис. 3. Подготовка тестового изображения

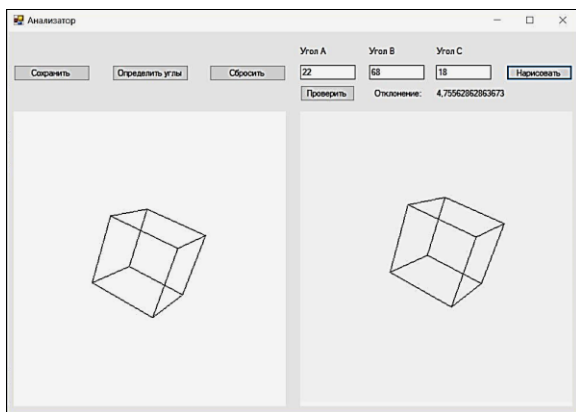


Рис. 4. Графическое отображение результата работы

Разработанный алгоритм был протестирован на изображениях с частичным перекрытием объекта. Для этого были созданы тестовые изображения, на которых часть объекта была удалена с помощью графического редактора, что создало эффект перекрытия.

На рис. 5 представлен один из примеров полученного тестового изображения и результат работы алгоритма соответственно.

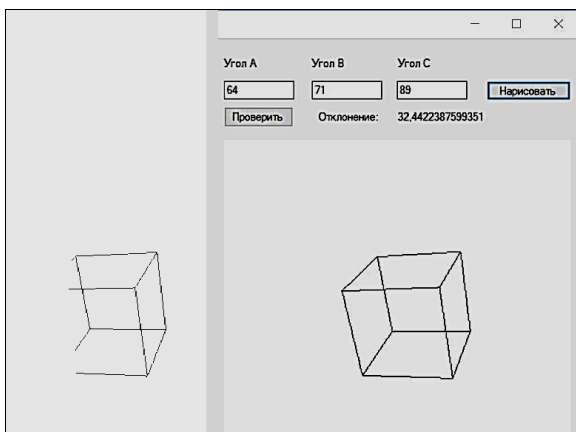


Рис. 5. Объект с перекрытием и соответствующий результат работы

Для демонстрации того, что разработанный алгоритм локализации применим к разным объектам, а не только к кубу, программно были заданы модели других объектов, а именно прямоугольного параллелепипеда и четырехугольной пирамиды. Тестирование показало, что разработанный алгоритм так же точно определяет углы поворота новых объектов. Результаты приведены на рис. 6 и 7.

Также работа алгоритма была протестирована на реальных изображениях, полученных с камеры мобильного телефона. В качестве объекта на изображении выступал куб, сцена была однородной. На рис. 8 представлен один из результатов работы алгоритма с реальным изображением.

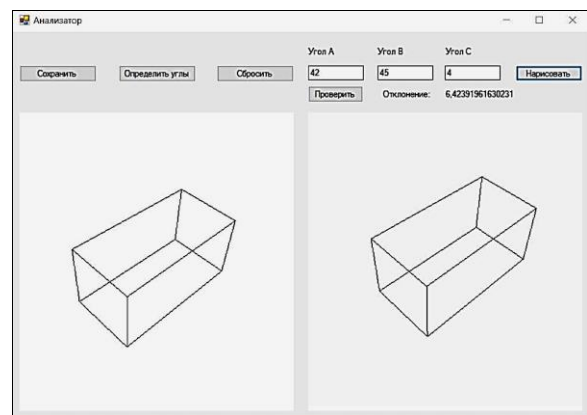


Рис. 6. Работа алгоритма с прямоугольным параллелепипедом

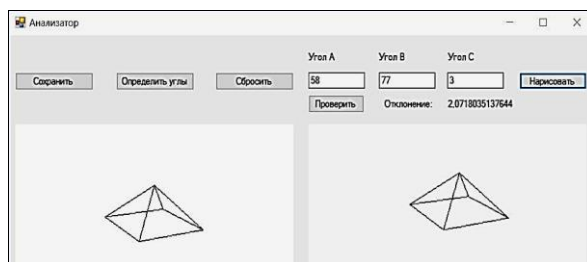


Рис. 7. Работа алгоритма с четырехугольной пирамидой

В целом на подготовленных тестовых изображениях было проведено несколько десятков тестов, на всех изображениях углы поворота были определены достаточно точно: среднее отклонение по каждой вершине для объекта без перекрытия не превышало 8 пикселей по совокупности координат, в среднем составляло 4 пикселей, а зрительно полученное изображение практически совпадало с анализируемым.

Данные результаты свидетельствуют о корректности разработанного подхода.

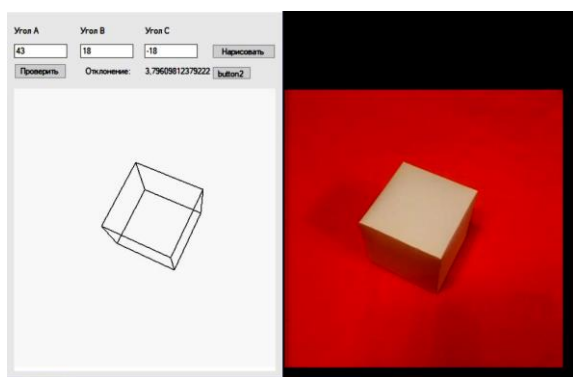


Рис. 8. Работа алгоритма с реальным изображением

Заключение

В результате данного исследования был проведен анализ существующих систем машинного зрения, а также алгоритмов и методов распознавания объектов на изображении. Был разработан оригинальный алгоритм локализации объекта, заданного трехмерной моделью, на изображении.

Созданный прототип приложения позволяет создавать тестовые изображения, а также определять углы поворота объекта на данном изображении. После выполнения алгоритма приложение графически отображает результат работы. Экспериментальная проверка результата показала, что разработанная система работает с высокой точностью.

К достоинствам разработанного алгоритма можно отнести устойчивость к частичному перекрытию объекта, когда объект на изображении виден не полностью.

В дальнейшем, прежде всего, следует обратить внимание на скорость работы алгоритма, а также способ задания модели.

Список литературы

1. *ГОСТ 25686-85*. Манипуляторы, автооператоры и промышленные роботы. Термины и определения: введ. 1986-01-01. М.: Госкомитет СССР по стандартам.
2. *Потанов А.* Системы компьютерного зрения. Современные задачи и методы // Международный журнал "Control Engineering Россия". 2014. № 1. С. 21–26.
3. *Системный интегратор в области измерений и автоматизации "ВиТэк"* URL: <http://www.vitec.ru/systems/sistemy-mashinnogo-zreniya-i-robototekhnika/> (дата обращения: 10.10.2017).
4. *Комплексная автоматизация технологических процессов, производства и предприятий ГК "ЭКСПРЕСС НТС-ЭКО"* URL: http://www.techrends.ru/realizedprojects/sistemy_tehnicheskogo_zreniya/ (дата обращения: 17.10.2017).
5. *COGNEX*. URL: <http://www.cognex.com/products/machine-vision/in-sight-vision-systems/> (дата обращения: 14.09.2017).
6. *Кудрина М.А.* Использование преобразования Хафа для обнаружения прямых линий и окружностей на изображении // Известия Самарского научного центра Российской академии наук. 2014. Т. 16. № 4–2. С. 476–478.
7. *Ту Дж., Гонсалес Р.* Принципы распознавания образов / пер. с англ. И.Б. Гуревича / под ред. Ю.И. Журавлева. 1978. С. 105–107.

Algorithms for objects precise localization based on visual sensors of industrial robots

A. Yu. Gorodilov, A. V. Shaburova

Perm State University; 15, Bukireva st., Perm, 614990, Russia
aleksey.gorodilov@gmail.com | alexshaburova@gmail.com

The article deals with the problem of localization of objects in the image, which occurs when using industrial robots. A method is proposed that makes it possible to identify the object and determine its position in space, regardless of the angle of the survey. The developed algorithm includes the Hough transformation to determine the potential vertices of the object, filtering out false candidates based on clustering methods, and optimized enumeration of variants of model and object image matching. As a result, a prototype of the system was created, which allows (on an example of a cube) to determine the angles of rotation of an object in space based on its image.

Keywords: *object localization; pattern recognition; Hough transformation; clustering; central projection; Euler angles.*