

Е. С. Ковалев, А. Ю. Беззубцев, А. В. Смирнов
**Разработка алгоритмов передвижения
квадрокоптера в узких помещениях.**

Аннотация. Разработка алгоритма полёта квадрокоптера в узких помещениях. Он должен лететь вдоль коридора (не обязательно прямого), избегая препятствий, без вмешательства человека.

Ключевые слова и фразы: Обработка изображений, программирование.

Введение

В настоящее время беспилотные летательные аппараты (БПЛА) завоёвывают всё больше места в самых различных областях. В силу своей специфики они решают огромный спектр задач при малых затратах. Развлечение, транспортировка грузов, фотографирование местности для картографии, видеосъёмка, уничтожение противника — лишь малая часть возможного применения «беспилотника». К сожалению, сам по себе аппарат может не так уж и много, в сложных ситуациях управление осуществляется человеком. Реализация полной автономности — одна из наиболее острых проблем в области БПЛА.

Задача

Разработать алгоритм перемещения БПЛА в узком помещении (коридоре). «Беспилотник» должен лететь вдоль него, избегая препятствий, без какого-либо вмешательства человека. Параметры коридора неизвестны, и летательный аппарат должен сам понять, куда ему нужно переместиться.

Разработка

Движение БПЛА выполняется по специальным ключевым точкам. Эти точки могут устанавливаться в зависимости от показателей фронтальной камеры, компаса, gps-навигатора, дальномера и иных встроенных датчиков.

© Е. С. Ковалев⁽¹⁾, А. Ю. Беззубцев⁽²⁾, А. В. Смирнов⁽³⁾ 2015

© УГП имени А. К. Айламазяна⁽¹⁾ 2015

© Институт программных систем имени А. К. Айламазяна РАН^(2, 3) 2015

В распоряжении имеется квадрокоптер, на котором установлена фронтальная камера. Квадрокоптер — это летательный аппарат с четырьмя несущими винтами. Для реализации алгоритма был выбран язык C++ и библиотека работы с компьютерным зрением OpenCV[1].

Для нахождения и фиксации ключевой точки, являющейся ориентиром для БПЛА, был разработан алгоритм «точка схода», цель которого — движение относительно центра коридора и предотвращение столкновений со стенами. Суть алгоритма сводится к тому, что на изображении производится поиск отрезков прямых линий и общей точки пересечения линий, фрагментами которых являются найденные ранее отрезки. Полученная точка является "точкой схода" и лежит на линии горизонта.

Структура алгоритма:

1) **Загрузка изображения.**

На рис.1. входное изображение.



Рис. 1. Входное изображение

2) **Фильтрация изображения для удаления шумов.**

Изображение сглаживается применением к нему «двухстороннего фильтра» [2], описываемого формулой:

$$I^{filtered}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_r) - I(x)\|) g_s(\|x_i - x\|)$$

Где термин нормализации:

$$W_p = \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|)g_s(\|x_i - x\|)$$

- $I^{filtered}$ отфильтрованное изображение;
 - I оригинал входного изображения;
 - x координаты текущего пикселя для фильтрации;
 - Ω окно, отцентрированное по x ;
 - f_r ядро диапазона сглаживания различий в интенсивности. Эта функция может быть функцией Гаусса;
 - g_s пространственное ядро для сглаживания различий в координатах. Эта функция может быть функцией Гаусса.
- На рис. 2. сглаженное изображение.



Рис. 2. Сглаженное изображение

3) Поиск на изображении краевых пикселей.

Края (границы) — это такие кривые на изображении, вдоль которых происходит резкое изменение яркости или других видов неоднородностей. Край — это резкий переход, изменение яркости. Причины возникновения краёв:

- изменение освещённости;
- изменение цвета;
- изменение глубины сцены (ориентация поверхности).

Выделение границ выполняется детектором границ Кенни[4].

Шаги детектора:

- убрать лишние детали из изображения;

- рассчитать лишние части изображения;
- сделать края тонкими (edgethinning);
- связать края в контура (edgetlinking).

Таким образом, получается двоичное изображение, содержащее границы (т.н. «тонкие края»). Результат работы детектора границ Кенни показан на рис. 3.

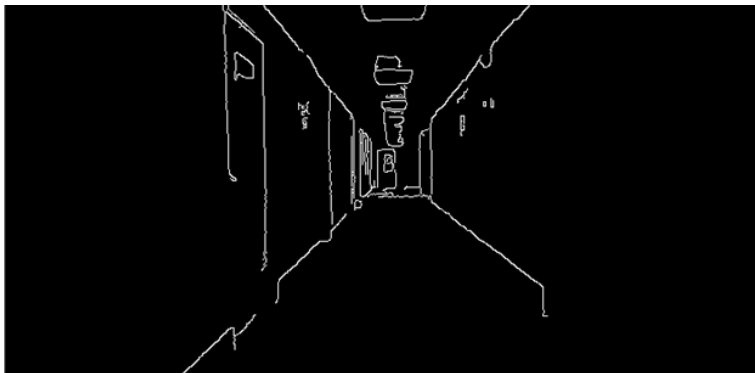


Рис. 3. Результат работы детектора границ Кенни

- 4) **Поиск на изображении отрезков прямых линий с помощью преобразования Хафа**[4]. В мире преобладают прямые линии и другие простые геометрические формы (квадрат, прямоугольник, треугольник, круг). Поэтому, одна из задач зрения робота — детектирование этих линий (для поиска каких-либо геометрических форм, например, дверного проёма, круглой розетки, коридора и т.п.) Эта задача успешно решается с помощью преобразования Хафа. Самым распространённым и простым случаем преобразования Хафа является поиск линий. В основе теории преобразования Хафа лежит утверждение, что любая точка двоичного изображения может быть частью некоторого набора возможных линий.

Формула линии:

$y = a * x + b$ в декартовых координатах

$p = x * \cos(f) + y * \sin(f)$ в полярных координатах

Прямую на плоскости можно представить:

$x * \cos(f) + y * \sin(f) = r$, где

r — длина перпендикуляра, опущенного на прямую из начала координат,

f — угол между перпендикуляром к прямой и осью OX . f находится в пределах от 0 до 2π ,

r ограничено размерами входного изображения.

Через каждую точку (x, y) изображения можно провести несколько прямых с разными R и f , то есть каждой точке (x, y) изображения соответствует набор точек в фазовом пространстве (R, f) , образующий синусоиду. Так же каждой точке (R_0, f_0) пространства (R, f) можно поставить в соответствие счетчик, соответствующий количеству точек (x, y) , лежащих на прямой

$$X * \cos(f_0) + y * \sin(f_0) = r_0$$

Теперь непрерывное фазовое пространство нужно перевести в дискретное, введя сетку на пространстве (R, f) , одной ячейке которой соответствует набор прямых с близкими значениями R и f .

Результат нахождения линий на изображении, с использованием преобразования Хафа показан на рис. 4.



Рис. 4. Результат преобразования Хафа

- 5) **Анализ угла наклона линий.** Если угол наклона линии к вертикали меньше 28 градусов или угол наклона к горизонтали меньше 12 градусов, то такая линия исключается. Линии, прошедшие отбор, сохраняются. Параметры линий будут использованы для отбора прямых имеющих точки пересечения.



Рис. 5. Результат сортировки линий

- 6) **Поиск координат точек пересечения прямых.** Если эти координаты не выходят за границы изображения, то сохраняем их в вектора. Если точка пересечения двух произвольных линий из набора находится в окне, то сохраняем координаты этой точки.
- 7) **Поиск координат «точек схода».** Каждая точка пересечения имеет свой вес, который определяет ее вклад в координаты «точки схода». Чем дальше точка от центра изображения тем меньше ее вклад. Вклад обратно пропорционален экспоненте, аргументом которой является расстояние от центра изображения.



Рис. 6. Точки с наибольшим весом



Рис. 7. «Точка схода»

Ниже приведены результаты работы алгоритма в различных коридорах и на разных высотах.

Полученная «точка схода» является ориентиром для БПЛА на всю продолжительность полёта по коридору. Движение к «точке схода» продолжается до тех пор, пока следующая ключевая точка не войдёт в поле зрения фронтальной камеры «беспилотника». Во время полёта возникала потеря «точки схода» и обнуление её координат. Это обуславливается наличием вибрации корпуса БПЛА и сопутствующим «эффектом желе». Решение данной проблемы заключается в сохранении предыдущих ненулевых координат «точки схода», до появления новых таких же ненулевых координат. Другими словами, БПЛА при потере точки, ориентируется по старым её координатам, пока вновь не обнаружит точку или маркер.

Представленный алгоритм показывал хорошие результаты на фотографиях различных коридоров. Но в ходе испытания на видеопотоке было выявлено несколько недостатков данного алгоритма:

- 1) Чрезвычайно низкая частота кадров, непригодная для использования в режиме реального времени — всего 4 кадра в секунду. Квадрокоптер вполне мог столкнуться со стеной прежде чем обнаружил бы это.
- 2) Расчёт «точки схода» производился с опорой на центр фронтальной камеры, что не отражало реального положения центра



Рис. 8. Результат работы алгоритма «точка схода»

коридора. В некоторых случаях отклонения между найденной и истинной точкой были недопустимо высоки.

- 3) Координаты «точки схода» меняются каждый кадр. Если расстояние между текущей и предыдущей точками больше сотни пикселей, курс будет меняться слишком резко. Неизвестно, как поведёт себя «беспилотник» в такой ситуации.
- 4) Алгоритм не предусматривал остановок.

При детальном рассмотрении алгоритма было выяснено, что значительную часть времени занимает операция фильтрации. Тестирование различных фильтров показало, можно получить результаты, схожие с результатами «двухстороннего фильтра», за меньший отрезок времени. Изображение сглаживается применением операции свертки с дискретным вариантом Гауссиана. Гауссиан [3] – математическая функция, описываемая следующей формулой:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

где μ и σ – вещественные числа.



Рис. 9. Результат применения гауссианы

Замена «двухстороннего фильтра» на гауссиану позволила достичь пиковой частоты 10 кадров, стабильной – 8 кадров, а при нагрузках 6 кадров в секунду.

Расчёт координат точки схода был изменён. Высчитывается среднее значение точек пересечения всех найденных линий. Таким образом, находится реальный центр коридора, независимо от того, насколько точно позиционирован БПЛА относительно него.

Быструю смену «точки схода» удалось устранить, высчитывая среднее значение последних найденных точек. Среднее значение меняется медленнее, чем «точка схода», но достаточно быстро, чтобы БПЛА мог вовремя позиционировать себя. Эвристически оптимальная скорость изменения ключевой точки достигалась при расчёте 6 последних точек.

Задача остановки «беспилотника» решается введением счётчика кадров, на которых точка не была обнаружена. Иными словами, если точка схода не находится в течение определённого промежутка времени, БПЛА останавливается. В дальнейшем планируется реализовать поворот квадрокоптера в непрямом коридоре. Для определения необходимости поворота так же можно использовать счётчик кадров. Скажем, если в течение определённого времени «точка схода» не будет обнаружена, квадрокоптер останавливается и выполняет поворот вокруг своей оси. При этом ключевые точки, захваченные между 165 и 195 градусами относительно изначального движения учитываться не будут, в противном случае «беспилотник» вернётся в исходную точку.

Заключение

Так как алгоритм работает с каждым кадром, нахождение новой ключевой точки не занимает много времени. Так же на обнаружение «точки схода» влияет однородность стен (в плане цвета) и освещённость помещения. В плохо освещённом коридоре определить координаты точки практически невозможно, из-за затруднений в поиске краевых пикселей на изображении. Чтобы изображение лучше «читалось» при плохом освещении можно установить на БПЛА камеру с инфракрасной подсветкой. Такие камеры появились относительно недавно, но уже успели приобрести популярность за счет широких возможностей видеонаблюдения и работе в крошечной тьме.

Список литературы

- [1] OpenCV [Электронный ресурс]: [официальный сайт].— <http://opencv.org/> ↑ 6.
- [2] Двухсторонний фильтр [Электронный ресурс] // Википедия: свободная энциклопедия. — 26 август. 2014.— http://en.wikipedia.org/wiki/Bilateral_filter ↑ 6.
- [3] Гауссова функция [Электронный ресурс] // Википедия: свободная энциклопедия. — 20 фев. 2015.— http://en.wikipedia.org/wiki/Gaussian_function ↑ 13.

- [4] OpenCV шаг за шагом [Электронный ресурс] // RoboCraft: [сайт] .— <http://robocraft.ru/page/opencv/> ↑ 7, 8.

Специфика статьи: Развитие информационно-вычислительных технологий, Развитие авиационно-космических технологий, Развитие робототехники и сложной механики, Алгоритм, Подпрограмма или библиотека программ, Средства компьютерной алгебры, Языки программирования, Вычислительный эксперимент, Методы теории сигналов и обработки изображений, Программы для разработки технических устройств.

Научный руководитель:

к.т.н. И. П. Тищенко

Об авторах:

Егор Сергеевич Ковалев

УГП имени А. К. Айламазяна, 4И11

e-mail:

Air1618@gmail.com

Артём Юрьевич Беззубцев

Институт программных систем имени А. К. Айламазяна РАН

e-mail:

mannaz2012@mail.ru

Александр Владимирович Смирнов

Институт программных систем имени А. К. Айламазяна РАН

e-mail:

asmirnov_1991@mail.ru

Пример ссылки на эту публикацию:

Е. С. Ковалев, А. Ю. Беззубцев, А. В. Смирнов. «Разработка алгоритмов передвижения квадрокоптера в узких помещениях». *Научоёмкие информационные технологии: Труды XIX Молодежной научно-практической конференции SIT-2015*. УГП имени А. К. Айламазяна. — Переславль-Залесский: Изд-во «Университет города Переславля», 2015 с. 5–16.

URL

<https://edu.botik.ru/proceedings/sit2015.pdf>

Egor Kovalev, Aleksandr Smirnov, Artem Bezzubcev. *Development of algorithms of quadrocopter movement in narrow location.*

ABSTRACT. Development of algorithms of quadrocopter flight in narrow location. It must moving along the corridor (it may be curved) avoiding obstacle without human interfering.

Key Words and Phrases: image processing, programming.

Sample citation of this publication:

Egor Kovalev, Aleksandr Smirnov, Artem Bezzubcev. “Development of algorithms of quadrocopter movement in narrow location”. *Science-intensive information technologies: Proceedings of XIX Junior R&D conference SIT-2015. Ailamazyan Pereslavl University.* — Pereslavl-Zalesskiy: Pereslavl University Publishing, 2015 pp. 5–16. (*In Russian.*)

URL

<https://edu.botik.ru/proceedings/sit2015.pdf>