

УДК 004.031.2, 004.932

Методы оптимизации вычисления оптического потока в автономных системах навигации

Жуков Р.В., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы обработки информации и управления»*

Научный руководитель: Терехов В.И., к.т.н, доцент

Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана

bauman@bmstu.ru

Введение

В автономных системах навигации очень важно иметь возможность определять перемещение объекта в пространстве. Для этого используются такие подходы как визуальная одометрия, позиционирование по спутникам (GPS, ГЛОНАСС), одометрия, основанная на вращении колес, использование инерциальных измерительных приборов. Каждый из этих методов обладает своими плюсами и минусами [1], но развитие вычислительной техники и алгоритмов компьютерного зрения дало мощный толчок к более широкому применению визуальной одометрии. Данный подход позволяет получать один видеоряд через видеокамеру и на его основе получать разные сведения об окружающей среде.

Постановка задачи и существующие решения

Задача определения перемещения на основе видео ряда на текущий момент, как правило, решается с использованием двух камер или одной стерео камеры [2]. На основе пары кадров, полученных с разных камер, строится стерео изображение, путем нахождения соответствия между пикселями. Найдя соответствие для каждого пикселя в виде вектора смещения, можно вычислить расстояние до объектов на сцене, при условии, что известно взаимное расположение камер и их параметры. Классическим методом решения данной задачи является алгоритм Stereo Block Matching (block-matching). Данный алгоритм достаточно хорошо изучен, подробная информация может быть найдена в работах [3], [4].

Сравнивая расстояния до объектов на последовательности кадров можно определять перемещение самой камеры, и, следовательно, самого объекта, на котором закреплена эта камера.

Однако такие камеры стоят порядка 200\$ и требуют специальных программных библиотек для работы с ними. Это накладывает ограничения на область применения стерео камер и систем, основанных на них.

Альтернативой является применение обычных бытовых камер и более сложных алгоритмов вычисления оптического видео потока, таких как *Farneback*, «простой метод» вычисления оптического потока и метод Лукаса-Канаде (*Lucas-Kanade*).

Непосредственное вычисление оптического потока не позволяет оценить перемещение объекта напрямую и на данный момент нет единого подхода к построению визуальной одометрии с его использованием, а высокая вычислительная сложность применяемых алгоритмов компьютерного зрения ограничивает применение такого подхода на встраиваемых и мобильных устройствах.

Предлагаемое решение

В статье предлагается решение задачи построения визуальной одометрии на основе одной бытовой камеры и относительно слабых вычислительных устройств. Такое решение требует соблюдения двух условий, которые приемлемы при создании большинства современных мобильных роботов:

- перемещение происходит по земле или другой горизонтальной плоскости;
- камера жестко закреплена относительно самого робота.

При соблюдении этих условий работу визуальной одометрии можно описать в 8 шагов [5].

- 1) Исправление изображения для исключения искажения линз.
- 2) Вычисление оптического потока для кадра.
- 3) Проверка полученных векторов смещения, исключение движущихся в кадре объектов, исключение ошибочных векторов.
- 4) Разделение всего оптического потока на две части: «наземную» и «небесную».
- 5) В «небесной» части перейти к цилиндрической системе координат и высчитать угол поворота относительно двух последних кадров, определяя тем самым угол поворота θ .
- 6) В «наземной» части выделить векторы $\{u, v\}$ из оптического потока и вычислить перемещение в плоскости $x-y$, получив вектор (x, y) .

- 7) Прибавить (x, y, θ) к изначальному положению объекта (X, Y, Θ) , получив новое положение объекта.
- 8) Перейти к шагу 1 для следующего кадра. Периодически обновлять ключевые точки.

В общем виде процесс вычисления оптического потока представлен на рисунке 1.

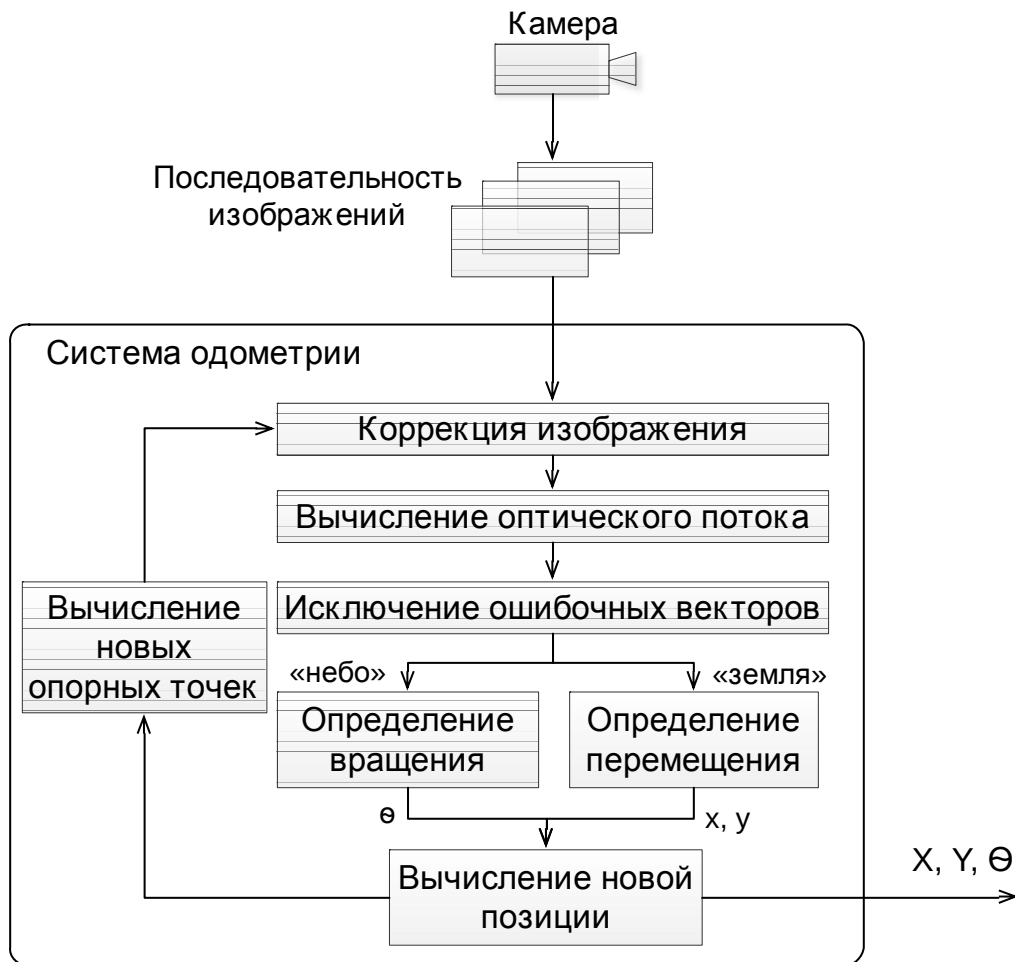


Рис. 1. Общая схема работы системы визуальной одометрии

Анализируя все шаги представленного алгоритма, можно выделить следующие трудоемкие задачи:

- вычисление оптического потока;
- выделение ключевых точек;
- выделение ошибочных векторов и исключение их из потока.

Для решения задачи вычисления оптического потока существует несколько методов, которые можно разделить на следующие группы:

- блочные методы, методы основанные на нахождении наиболее похожей точки в окрестности, называемой окном, на двух последовательных кадрах. Одним из

представителей этой группы является простой алгоритм вычисления оптического потока SimpleFlow;

- дифференциальные, в основе которых лежит вычисление производной по горизонтали и вертикали для последовательности пикселей и нахождения наиболее близких значений для них. Данная группа представлена такими методами как метод Лукаса-Канаде, Фарнебэк, *CLG*;

- на основе дискретной оптимизации основаны на сложных вычислениях и методах линейного программирования. Известных устоявшихся подходов пока нет, так как данные методы являются относительно новыми и самыми ресурсоемкими. Так же отсутствуют готовые реализации методов из данной группы.

Производительность наиболее популярных методов вычисления оптического потока представлена на графиках, приведенных на рисунке 2.

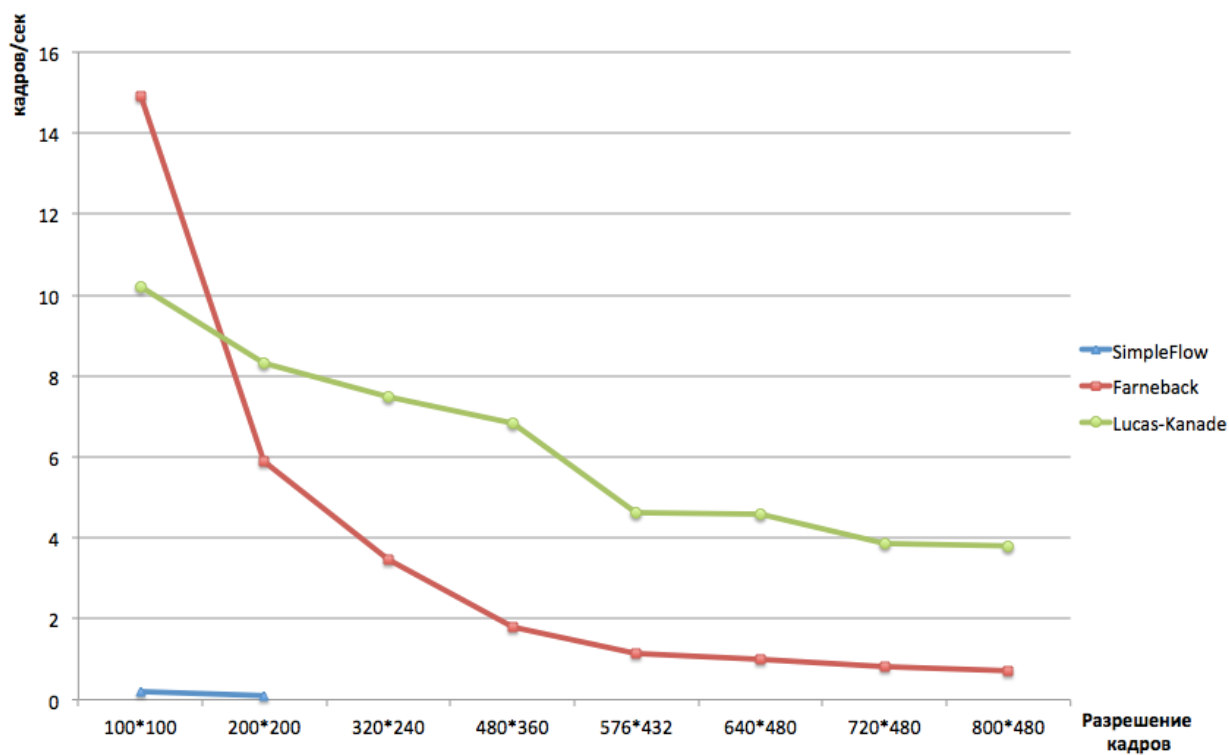


Рис. 2. Сравнение производительности алгоритмов вычисления оптического потока

Данные графики отображают возможную скорость обработки кадров в зависимости от их размера. Измерения производились на устройстве, которое относится к классу бытовых, что обусловлено постановкой задачи, и обладает 4-х ядерным процессором с тактовой частотой 1500 МГц, 2 ГБ оперативной памяти, видеокамеру с разрешением 8 Мпикс.

Из графиков видно, что при разрешении 640*480 пикселей, которое является приемлемым для решения поставленной задачи лишь метод Лукаса-Канаде позволяет обрабатывать порядка 5 кадров в секунду. Данное значение не является высоким, но оно все же позволяет решать задачу вычисления оптического потока в реальном времени.

Данный алгоритм основан на предположении, что пиксели на двух последовательных изображениях не меняют своего цвета, а лишь изменяют свое положение, что можно представить следующим образом [6]:

$$I(x, y, t) = I(x + u_x, y + u_y, t + 1),$$

где I - это функция яркости пикселей от положения на кадре и времени. Другими словами x и y - это координаты пикселя в плоскости кадра, U_x и U_y - это смещение, а t - это номер кадра в последовательности. При этом следует учитывать, что между двумя соседними кадрами проходит единичный отрезок времени [7].

Следует отметить, что данный метод находит смещение лишь определенных точек, а не всех пикселей изображения, при этом добавляется задача нахождения этих точек.

Конечно, в реальной обстановке от кадра к кадру могут меняться глобальные условия освещения и освещенность самого движущегося объекта. С этим допущением связана масса проблем, но, как ни странно, вопреки всему оно достаточно хорошо работает на практике.

Для оптимизации этого алгоритма применяется метод пирамид: изображение последовательно уменьшается в размерах в несколько раз и оптический поток вычисляется сначала для самого маленького изображения, получая общее, грубое смещение. Затем оптический поток уточняется на все больших копиях кадра, учитывая результаты предыдущего шага. Тем самым, несмотря на увеличение числа шагов, снижается вычислительная сложность каждого шага и общая вычислительная сложность. Данный алгоритм эффективно реализован в открытой библиотеке компьютерного зрения OpenCV [8].

Оптимизировать выделение ошибочных векторов из оптического потока можно лишь на основе построения гистограммы векторов и исключения сильно выделяющиеся. Так же следует учитывать значения на предыдущем шаге, чтобы определять движущиеся объекты в кадре. Какие-то специальные приемы по снижению трудоемкости этой задачи не известны.

Значительное увеличение производительности алгоритма можно получить при оптимизации поиска ключевых точек. Одним из методов оптимизации такого поиска

является метод сохранения ключевых точек между кадрами, без их покадрового обновления.

Так как поиск ключевых точек на кадре сам по себе является довольно трудоемкой задачей, то исключение этого этапа на каждом цикле работы алгоритма позволяет значительно сократить общую трудоемкость процесса визуальной одометрии. Однако при этом возникает вопрос выбора интервала обновления опорных точек. При этом может возникнуть ситуация, при которой на одном кадре ключевых точек будет крайне мало, например при резкой вспышке света. Если обновление точек производить как раз в этот момент, то до следующего обновления результат работы алгоритма будет крайне неточен.

Для решения этой проблемы предлагается ввести вес точки, который будет изменяться от кадра к кадру, а при достижении определенной величины, точка будет удалена, а вместо нее будет найдена другая.

Изменение веса может происходить следующим образом:

- если эта ключевая точка сдвинулась относительно двух кадров, то ее вес уменьшается на q ;
- если эта ключевая точка не сдвинулась, ее вес увеличивается на w ;
- при достижении порога в e , точка удаляется.

Значения $q/w/e$ необходимо подбирать исходя из особенностей окружающей среды, но наиболее общими можно считать значения 1/5/10. Такие значения сокращают частоту обновления контрольных точек как минимум в 2 раза при статичной сцене. Однако вопрос выбора весов является предметом отдельного исследования.

Другим методом оптимизации вычисления оптического потока является разделение изображения на области, в каждой из которых будет найдена только одна ключевая точка. Данный подход позволяет не тратить вычислительные мощности на «пестрые» объекты в кадре, а равномерно распределить ключевые точки по всему кадру.

Еще одним способом снижения вычислительной сложности описанного выше алгоритма является ввод лимита на число ключевых точек на кадре. В совокупности с предыдущими двумя подходами данный позволит гибко увеличивать скорость выполнения алгоритма, уменьшая при этом точность. Однако зависимость точности от числа ключевых точек имеет не линейную, а скорее логарифмическую зависимость, поэтому их бесконечное увеличение не имеет смысла. Поэтому введение ограничения на их общее число может так же благоприятно сказаться на быстродействии системы визуальной одометрии.

Заключение

В настоящий момент происходит снижение цен на видеокамеры, причем гораздо быстрее, чем на другие сенсоры. На практике даже единичные камеры могут быть применены в построении мобильных платформ, так как через камеру может быть получен большой объем различных данных об окружающем мире. При этом текущие алгоритмы компьютерного зрения по обработке изображений и видео потока далеки от совершенства и требуют серьезных вычислительных мощностей. Даже построение системы с компьютерным зрением на открытых и отлаженных библиотеках не позволяет применять эти алгоритмы на мобильных и автономных устройствах. Статья описывает систему визуальной одометрии на базе одной камеры, а также предлагает несколько подходов к оптимизации базового алгоритма, которые позволяют снизить требования к аппаратному обеспечению.

Список литературы

1. Жуков Р.В. Методы оценки положения объекта в пространстве // Наука и образование: МГТУ им. Н.Э. Баумана Электрон. журн. 2012. № 10. Режим доступа: <http://sntbul.bmstu.ru/doc/636938.html> (дата обращения 5.11.2013).
2. Scaramuzza D., Fraundorfer F., Siegwart R., Real-Time Monocular Visual Odometry for On-Road Vehicles with 1-Point RANSAC // Robotics and Automation. ICRA '09. IEEE International Conference (Anchorage, May, 12-17, 2009): proceedings. Anchorage, 2009. P. 4293 – 4299.
3. Bradski G., Kaegler A. Computer Vision with the OpenCV library // IEEE International Conference ICRA '10 «Robotics and Automation» (Paris, 5-18, May, 2010): proceedings. Paris, 2010. P. 3564 – 3569.
4. Scharstein D., Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms // International Journal of Computer Vision. No. 47(1/2/3). P. 7-42.
5. Shude Guo, Cai Meng. Monocular Visual Odometry and Obstacle Detection System Based on Ground Constraints. Berlin Heidelberg: Springer, 2012. 525 p.
6. Алгоритм Лукаса-Канаде. Свободная энциклопедия Википедия. Режим доступа: http://en.wikipedia.org/wiki/Lucas%E2%80%93Kanade_method (дата обращения 17.11.2013).
7. Вычисление оптического потока методом Лукаса-Канаде. Теория. Коллективный блог habrahabr. Режим доступа: <http://habrahabr.ru/post/169055/> (дата обращения 25.12.2013)

8. Motion Analysis and Object Tracking. Документация к проекту OpenCV. Режим доступа:
http://docs.opencv.org/modules/video/doc/motion_analysis_and_object_tracking.html (дата обращения 13.01.2014).