Адаптивный подход к распознаванию объектов с произвольного ракурса в реальном времени

Алексей Мастов[1,2], Иван Коноваленко[1,2], Антон Григорьев[1,2]

[1] Институт проблем передачи информации РАН
[2] Московский физико-технический институт (Государственный университет)
mastov@phystech.edu

Аннотация Задача распознавания объектов с произвольного ракурса в реальном времени рассмотрена в рамках проекта по созданию автономного наземного робота. Для ее решения был использован алгоритм random ferns, основанный на парадигме особых точек с адаптивным выбором дескрипторов. В данной статье приводится описание адаптации этого алгоритма к решению задачи поиска объектов в видеопотоке в реальном времени.
Исследование выполнено в ИППИ РАН за счет гранта Российского научного фонда (проект № 14-50-00150).

Ключевые слова: распознавание образов, поиск объектов, метод особых точек, random ferns

1 Введение

Задача поиска объектов всегда была одной из самых популярных в области компьютерного зрения. Кроме этого, интерес к ней в последнее время подкрепляется еще и увеличивающимся спросом со стороны робототехники. Для распознающих систем, используемых в роботах, важна работа в реальном времени (20–30 кадров в секунду), а также устойчивость к изменениям условий наблюдения и положения объекта. Одним из возможных алгоритмов, удовлетворяющих вышеуказанным требованиям является Random Ferns [1], который относится к семейству методов, основанных на особых точках. Самым важным отличием этого алгоритма является замена трудоемкой оптимизационной задачи для сопоставления особых точек на задачу их классификации.

На сегодняшний день метод особых точек является мощнейшим инструментом, который используется не только для решения задач детектирования, но и калибровки камеры, 3d реконструкции, оценки собственного движения (egomotion estimation), создания панорам.

Согласно [2], началом использования данного метода можно считать работу Моравека [3] по стереосопоставлению. Но подход, который стал основой современных методов, был предложен Шмидом и Мором [4] в 1997.

Основная идея метода особых точек, как известно, заключается в поиске таких локальных особенностей на изображении, которые будут устойчивы

к целой серии преобразований (повороты, изменение масштаба, яркости и пр.). Поиск объекта по таким особенностям (а не как целого) позволяет эффективно бороться с сильными заслонениями объекта. Но для корректной работы алгоритм, очевидно, должен иметь возможность сопоставить эти особенности между собой как можно лучше. Поэтому можно выделить следующие требования к особым точкам:

- *Повторяемость (repeatability).* На двух разных изображениях одного и того же объекта должны быть найдены одинаковые особенности.
- *Различимость (distinctivity).* Разные особые точки должны быть отличимы друг от друга.

Одной из особенностей метода особых точек является то, что он включает в себя несколько этапов, каждый из которых может решаться независимо. Это многоэтапное построение алгоритма привело к большой вариации методов, предлагающих как комплексные решения, так и решения для каждого шага в отдельности. Стандартным разбиением алгоритма на шаги является следующее:

1. Нахождение особых точек (keypoints) с помощью детектора;
2. Описание точек дескриптором;
3. Сопоставление особых точек
4. Нахождение такого преобразования изображения, которое совместило бы данные точки.

Почти все используемые алгоритмы отличаются друг от друга только на первых двух этапах (детектирование и описание особых точек), в то время как уже стало общепринятым решать задачу сопоставления трудоемким поиском ближайшего соседа в какой-либо метрике. Алгоритм, который используется в данном исследовании, предлагает совершенно другой способ сопоставления особых точек. А именно, вместо оптимизационной задачи решается на задача классификации [1]. Кроме этого, такой подход делает возможным самообучение детектора.

В ходе настоящего исследования был разработан и протестирован модуль, позволяющий роботу находить определенные заранее объекты в видеопотоке и вычислять их относительные координаты.

1.1 Постановка задачи

Пусть имеется множество объектов $O_1, \ldots, O_n$. Каждому из них соответствуют их «эталонные» изображения (эталоны) с разных ракурсов — O_{ij}.

Задачу в таком случае можно поставить следующим образом: в каждый момент времени алгоритм должен находить в кадре все вхождения изображений из множества O_{ij} и, основываясь на полученных координатах, достраивать полную трёхмерную модель обнаруженных объектов O_i. Важно, что все эти действия должны происходить в реальном времени.

Также необходимо обеспечить устойчивость алгоритма к следующим особенностям наблюдения объекта в реальном мире:

– Изменение масштаба объекта, его поворот;
– Частичное заслонение искомого объекта;
– Произольное изменения условий наблюдения (освещения, тени, осадки);
– Неоднородность фона.

В рамках данного исследования множеством объектов являются параллелепипеды (коробки) различного вида и размера, а «эталонами» — изображения (фотографии) их граней.

2 Алгоритм Random Ferns

Одной из особенностей нашей задачи является возможность работы алгоритма в оффлайн режиме, например, для обучения. Это возможно благодаря тому, что искомый объект известен заранее и мы можем затратить неограниченое количество времени (в разумных пределах) на подобный этап "предподсчёта".

Использование этой возможности для заметного сокращения времени работы отражено в статье [1]. Ниже этот алгоритм, получивший название *Random ferns* (RFs), рассмотрен подробнее.

Постановка задача классификации Обозначим особые точки, найденные на объекте как $K_o = \{k_{o_1}, \cdots, k_{o_H}\}$, а $f_1, \cdots, f_N$ — бинарные признаки, описывающие каждую из них аналогично [5].

Сопоставим каждой точке из K_o класс $c_1, \cdots, c_H$. Таким образом, наша задача теперь заключается в том, чтобы каждой особой точке сцены сопоставить наиболее подходящий класс. Формально такой класс определяется как:

$$\widehat{c}_i = \arg\max_{c_i} P(C = c_i | f_1, \cdots, f_N),$$

где C — случайная величина, обозначающая класс особой точки сцены. Используя формулу Байеса можно перейти к

$$\widehat{c}_i = \arg\max_{c_i} P(f_1, \cdots, f_N | C = c_i). \quad (1)$$

Группировка признаков Поскольку признаки f_i простые (сравнение яркостей двух точек в окрестности особой), специфичности описания распознаваемого объекта можно достичь большим их количеством ($N \approx 300$).

Без каких-либо дополнительных предположений для оценки полного совместного распределения в (1) необходимо вычислить 2^N значений для каждого из классов, что предполагается слишком затратным по времени. С другой стороны, если предположить полную независимость признаков, вероятность в правой части (1) распадется в произведение $\prod_{j=1}^{N} P(f_j | C = c_i)$. Т.к. при таком допущении мы теряем много полезной информации о корреляции признаков, его использование также невозможно.

Найти «золотую середину» можно разбив все признаки на M групп по $S = \frac{N}{M}$, и считать независимыми эти группы. Такие блоки и были названы авторами *"фернами" (ferns)*.

Теперь условное распределение запишется как:

$$P(f_1, \cdots, f_N | C = c_i) = \prod_{j=1}^{M} P(F_j | C = c_i), \tag{2}$$

где F_j — множество из S случайно выбранных f_i. Т.к. F_j является двоичным вектором, ему можно однозначно сопоставить целое число $k \in [0, 2^S]$. Это свойство пригодится нам в дальнейшем для удобства обозначений.

Обучение Этап обучения предполагает оценку вероятностей в правой части (2). Т.е. нужно найти

$$p_{k,c_i} = P(F_m = k | C = c_i).$$

Обучение начинается с поиска наиболее стабильных особых точек на объекте. Для этого детектор запускается на деформированных копиях исходной картинки. Точки, обнаруженные чаще других, считаются стабильными, после чего им присваивается уникальный класс.

Следующим шагом обучения является создание обучающей выборки (Рис.1). Делается это путем многократных применений псевдослучайных (из заданного распределения) аффинных преобразований к исходному изображению.

Имея обучающую выборку, мы можем выполнить необходимую оценку p_{k,c_i} как:

$$p_{k,c_i} = \frac{N_{k,c_i} + 1}{N_{c_i} + 2^S}, \tag{3}$$

где N_{c_i} — количество элементов выборки, на которых присутствует точка класса c_i, и N_{k,c_i} — количество элементов выборки, на которых класс c_i описан k-тым ферном.

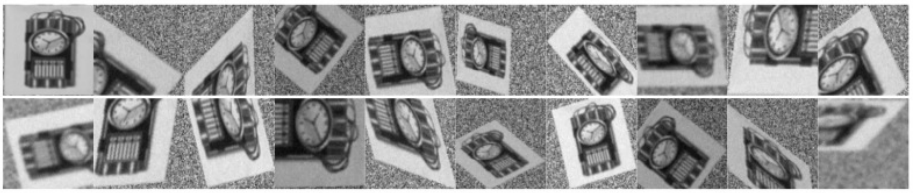

Рис. 1. Пример сгенерированной обучающей выборки. Первое изображение — эталон.

3 Адаптация Random Ferns к роботу

3.1 Входные данные

Несмотря на то, что робот обладает двумя камерами, модуль детекции использует монокулярное зрение и запускается параллельно для двух камер.

- На вход алгоритма поступает изображение сцены размером 1280×960, которое предварительно прошло через модуль исправления радиальной дисторсии;
- Матрица калибровки камеры $\boldsymbol{K}$ считается известной;
- Фотографии граней объекта (эталоны) хранятся на жестком диске робота и передаются узлу через запускаемый файл (Рис.2);
- Для каждой грани известно расстояние до противоположной (глубина).

Рис. 2. Пример эталонных изображений.

3.2 Описание модуля детектирования

При первом запуске распознающего модуля алгоритм обучается на эталонных изображениях и сохраняет необходимые параметры на жесткий диск. При повторном запуске обучение не требуется.

После обнаружения грани узел поиска сообщает координаты объекта в пикселях и расстояние до объекта в метрах узлу, отвечающему за движение к объекту. Одновременно с этим происходит вычисление координат противоположной грани. Имея координаты передней и задней граней, полная трехмерная модель коробки может быть однозначно восстановлена.

3.3 Детектирование нескольких граней

Выше для упрощения рассуждений предполагалось, что поиск объекта осуществляется по одной его грани. Естественным кажется обучение детекторов на каждой из k граней объекта и их параллельный запуск на каждом кадре. Ясно, что такое наивное решение замедляет работу в k раз.

Недостатки этого подхода очевидны:

1. Алгоритм k раз выполняет общий для всех детекторов шаг — поиск особых точек на сцене. Данная процедура в случае работы одного детектора занимает $\approx 30\%$ времени обработки кадра;
2. Выполняется поиск всех граней одновременно. Хотя вполне разумным является предположение, что на последующих кадрах в течение определенного времени перед камерой все еще будет располагаться грань, найденная на некущем кадре.

Мы избавились от указанных выше недостатков с помощью централизованного поиска особых точек сцены для всех детекторов и создания *очереди детекторов*.

Очередь детекторов Произвольно упорядочим k обученных детекторов $d_1, \dots d_k$. Пока не найдена какая либо из граней, каждый детектор в порядке очереди пытается найти ту, на которой он обучен. Если текущий детектор d_i нашел грань в кадре t, то на нем поиск заканчивается, и d_i становится первым в очередь детекции на кадре $t+1$. В противном случае право детектирования на кадре t переходит к d_{i+1}.

Таким образом, когда объект (какая-либо из его граней) находится в кадре, очередь детекторов работает по времени в точности как один. Если же объекта нет, согласно экспериментам, время работы увеличивается в $k/2$ раз.

3.4 Увеличение максимального расстояния детекции

Было проведено экспериментальное исследование описанного алгоритма на примере распознавания кубического модельного предмета с характерным размером 33 см. Было выяснено, что стандартный алгоритм очень плохо справляется с детектированием объекта, когда его размеры становятся в два раза меньше, чем при обучении. На практике это приводит к потере объекта уже на расстоянии около 2.5 метров.

В связи с этим мы предлагаем два способа увеличения максимального расстояния детекции:

- Вспомогательный детектор;
- Обучение одного детектора на нескольких изображениях.

Вспомогательный детектор Использование вспомогательного детектора является частным случаем одновременного поиска нескольких граней. Только теперь для каждой грани мы обучаем два детектора на разных ее размерах. Такой подход, очевидно, увеличивает время обработки кадра в два раза (в случае отсутствия объекта в кадре, когда это некритично), но увеличивает предельное расстояние детекции, по результатам эксперимента, на 1 метр.

Обучение на нескольких изображениях При постановке задачи не было ограничения на количество доступных фотографий объекта, но в стандартной реализации алгоритма используется только одна. Идея данного метода заключается в том, чтобы добавить в обучающую выборку реальных искаженных изображений объекта (изменение поворот, изменение масштаба и освещения) и затем их, так же как и исходное, подвергнуть многочисленным деформациям.

Данный метод можно реализовать без значительных изменений в структуре программы. Необходимо лишь знать проективные преобразования между новыми изображениями и исходным. Эти преобразования позволят нам вычислить новые координаты особых точек, избежав обычного детектирования, которое может найти отличное от исходного множество особых точек. Далее уже знакомый процесс обучения определяет необходимые вероятности.

В настоящее время не было проведено исследований, позволяющих сделать вывод об увеличении предельного расстояния детекции, но нами была выдвинута гипотеза, что такой подход способен его увеличить без существенного снижения производительности.

4 Заключение

Результатом данной работы является модуль поиска объектов, успешно внедренный в робота под управлением Robot Operating System. Некоторые примеры его работы приведены на Рис.3.

Данный модуль позволяет находить в реальном времени (характерное время обработки кадра 35 мс) объекты с характерным размером 33 см на расстояниях до 3.5 метров и обеспечивает устойчивость к отклонению положения грани от фронтального в пределах $[-\frac{\pi}{4}, \frac{\pi}{4}]$.

К перспективным направлениям данного исследования можно отнести увеличение предельного расстояния детекции до 5 метров, а также использование самообучения для улучшения характеристик классификатора.

Список литературы

1. Ozuysal, M., Calonder, M., Lepetit, V., Fua, P.: Fast Keypoint Recognition Using Random Ferns. IEEE Transactions on Pattern Analysis and Machine Intelligence (2010)

Рис. 3. Результаты работы алгоритма. Найденный объект выделен черным контуром. Помимо обнаруженной грани, на рисунке отображена восстановленная трехмерная модель объекта.

2. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints 60(2), 91–110 (2004)
3. Moravec, H.P.: Rover visual obstacle avoidance. Proc IJCAI pp. 785–790 (1981)
4. Schmid, C. and Mohr, R.: Local grayvalue invariants for image retrieval. Pattern Analysis and Machine Intelligence, IEEE Transactions on 19, 530–535 (1997)
5. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF : Binary Robust Independent Elementary Features. Proceedings of the European Conference on Computer Vision. ECCV'10 pp. 778–792 (2010)